

Analog Expansion Module

EX24



คุณสมบัติพิเศษ (Features)

- 16 ช่อง Analog Input เลือกชนิดสัญญาณได้ (Programmable Analog Input) 13 ชนิด ต่อ 1 channel
- แยกสัญญาณเข้า (Isolation) Analog ด้วย Relay, Digital ด้วย OPTO ELECTRONICS

อัตราการใช้งานสูงสุด (Absolute Maximum Ratings)

Analog Input ขั้ว 1 ถึง 2	(-) 0.3 – 15 V
Operating Temperature	5–50 °C

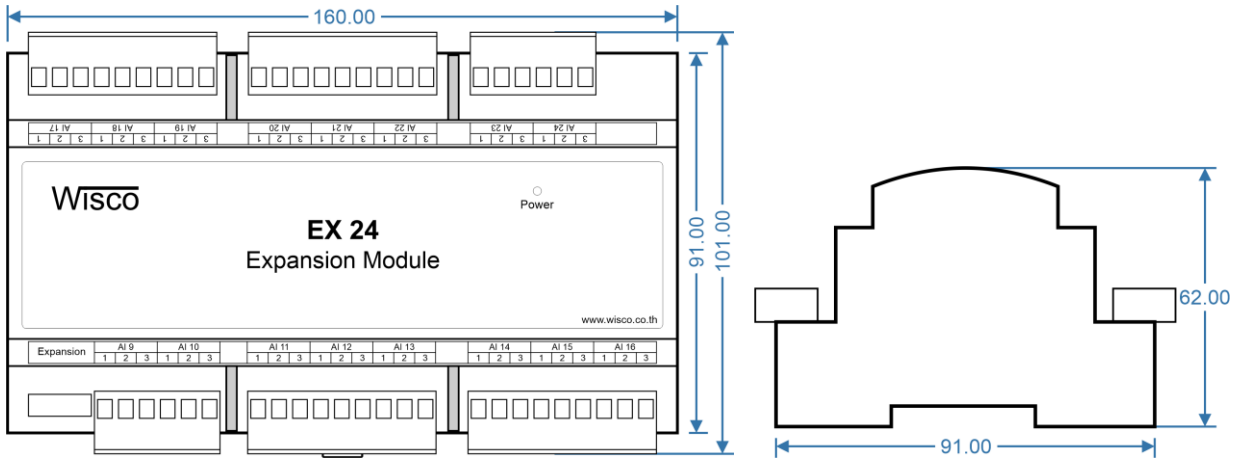
ข้อมูลเฉพาะ (Specifications)

Acquisition Time สูงสุด 0.3 วินาที (300 ms)

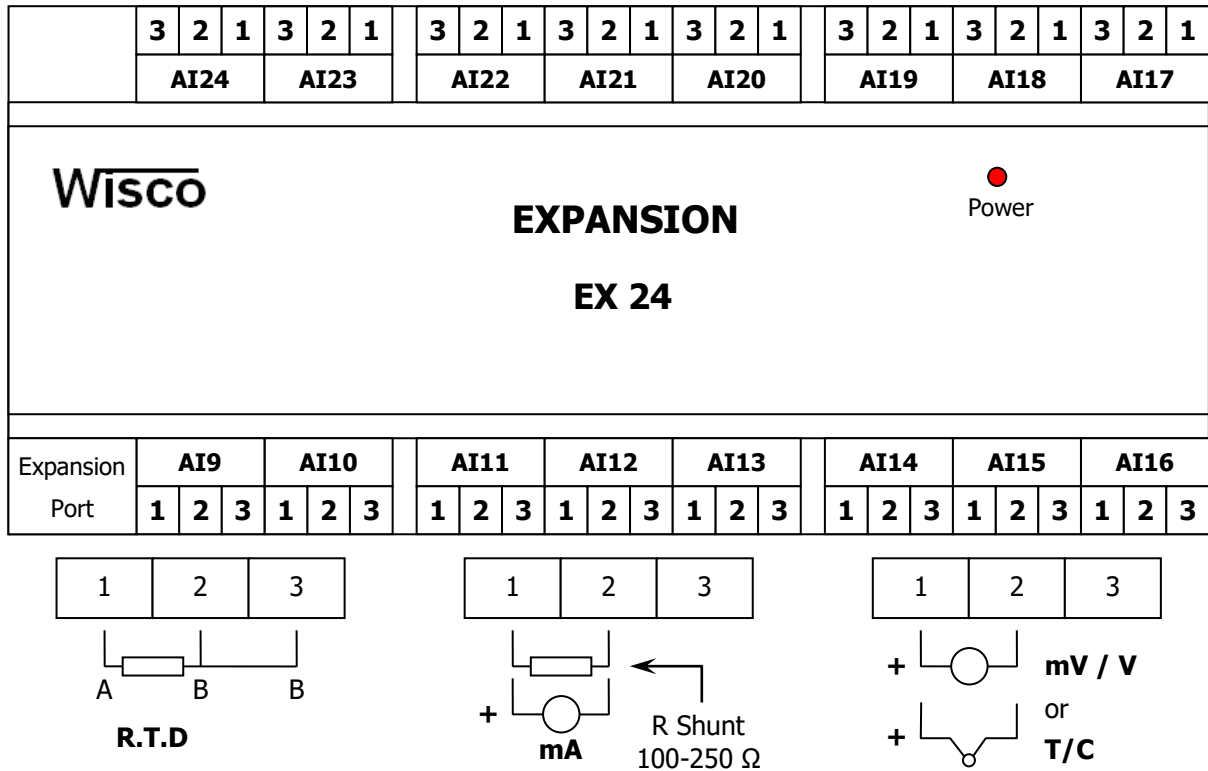
Analog Input

Resolution	ดูตารางในหน้าที่ 5
Accuracy	ดูตารางในหน้าที่ 5
Thermocouple	R, S, K, E, J, T, B
Cold Junction Temperature Compensation Accuracy	± 2 °C
RTD	Pt100 3-wire type
Amperage	≈ 0.25 mA
Resistance	สูงสุด 5 Ω ต่อ 1 สาย (5 Ω Max./Wire)
Voltage	0-10 V, 0-5 V, 0-100 mV
Input Impedance	≈ 200 kΩ
Current	0-20 mA, 0-40 mA (ต้องต่อความต้านทานภายนอกด้วย)
External Resistor	≈ 100-250 Ω

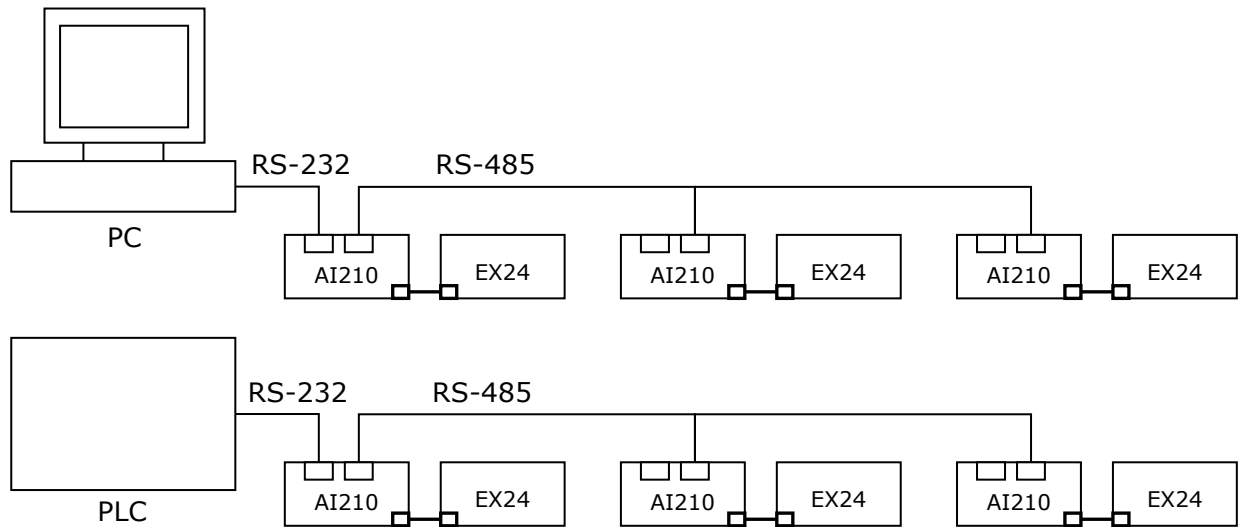
Dimension (Unit: mm.)



การต่อสาย (Wiring Diagram)



ตัวอย่างการใช้งานโมดูลตามมาตรฐาน RS-232/RS-485



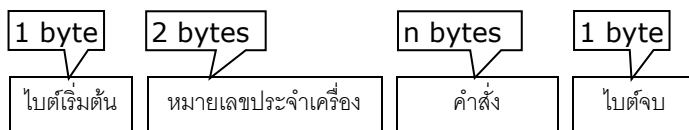
รหัสและย่านการวัดของช่องสัญญาณ Analog แต่ละชนิด

Code	Input Type	Measuring Range	Resolution	Accuracy (%FS) @25 °C	
00	Not Use	-	-	-	
01	Thermocouple	R	0 - 1700 °C	± 0.2% (3.4 °C)	
02		S	0 - 1700 °C	± 0.2% (3.4 °C)	
03		K	(-)250.0 - 1300.0 °C	0.1 °C	± 0.2% (2.6 °C)
04		E	0.0 - 1000.0 °C	0.1 °C	± 0.2% (2.0 °C)
05		J	(-)200.0 - 700.0 °C	0.1 °C	± 0.2% (1.4 °C)
06		T	(-)250.0 - 400.0 °C	0.1 °C	± 0.2% (0.8 °C)
07		B	0 - 1800 °C	1 °C	± 0.2% (3.6 °C)
08	R.T.D. Pt100	(-)200.0 - 800.0 °C	0.1 °C	± 0.2% (1.6 °C)	
09	Voltage(mV) 0 - 100	0.00 - 100.00 mV	0.01 mV	± 0.02% (0.02 mV)	
10	Voltage (V)	0 - 5	0.000 - 5.000 V	0.001 V	± 0.04% (0.002 V)
11		0 - 10	0.000 - 10.000 V	0.001 V	± 0.02% (0.002 V)
12	Current (mA)	0 - 20	0.00 - 20.00 mA	0.01 mA	± 0.1% (0.02 mA)
13		0 - 40	0.00 - 40.00 mA	0.01 mA	± 0.05% (0.02 mA)

EX24 จะใช้งานได้ต่อเมื่อต่อร่วมกับ **AI210** เท่านั้น ทั้งนี้มาตรฐานการติดต่อจะขึ้นอยู่กับ **AI210** ซึ่งมีอยู่สองมาตรฐานคือมาตรฐาน RS-232 และ RS-485 โดยทั้งสองมาตรฐานจะใช้ข้อกำหนด (Protocol) เดียวกันในการติดต่อกับ **EX24** โดยมีรายละเอียดดังต่อไปนี้

การติดต่อกับโมดูลโดยใช้ Wisco Protocol

ข้อมูลที่ใช้ในการติดต่อกับ **EX24** ผ่าน **AI210** จะเป็นรหัส ASCII ทั้งหมดและในคำสั่งชุดหนึ่งจะประกอบไปด้วย



ไบต์เริ่มต้น ไบต์แรกที่บอกให้โมดูลรู้ว่าได้เริ่มต้นของชุดคำสั่ง โดยจะใช้อักขระ '#' เป็นตัวเริ่มต้น

หมายเลขประจำเครื่อง หมายเลขที่ใช้อ้างอิงตัวโมดูลสำหรับกรณีที่มีการต่อใช้งานพร้อมกันตั้งแต่ 2 ตัว ขึ้นไป โดยสามารถตั้งได้ที่ DIP Switch บนตัวโมดูล ซึ่งจะมีค่าตั้งแต่ 00h-1Fh และห้ามให้หมายเลขซ้ำกัน

คำสั่ง คำสั่งที่ใช้กับ **EX24** จะมีทั้งหมด 8 คำสั่ง

ไบต์จบ ไบต์สุดท้ายที่บอกให้โมดูลรู้ว่าสิ้นสุดของชุดคำสั่ง โดยจะใช้ [CR] (Carriage Return) ซึ่งเป็นอักขระตัวที่ 13 ในตาราง ASCII เป็นตัวปิดท้าย

Character	#	0	0	R	A	I	1	2	4	5	8	CR
ASCII Code	23H	30H	30H	52H	41H	2AH	31H	32H	34H	35H	38H	0DH

ตัวอย่างการใช้งานคำสั่งสำหรับ Wisco Protocol

รายละเอียดและตัวอย่างของคำสั่ง **Wisco Protocol**

(= 1 byte, = n bytes, = Carriage Return)

1. คำสั่งที่ใช้อ่านค่า Analog Input

เริ่มต้นคำสั่งด้วย 'RAIX' ตามด้วยช่องสัญญาณที่จะอ่าน โดยใช้รูปแบบของ bit ทั้งหมด 6 ไบต์ (MSB -> LSB, '0' = ไม่อ่าน, '1' = อ่าน) และจบด้วย '[CR]' เช่น อ่านค่า AI จากเครื่องหมายเลข 02 ช่องที่ 24, 22, 20, 17, 16, 15, 10, 7, 4, 3, 2, 1 จะได้คำสั่งดังนี้ '#02RAIXA9C24F[CR]'

#	0	2	R	A	I	X	A	9	C	2	4	F	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	------

โดยตัวโมดูลจะตอบกลับมาเป็น 'AI>' ตามด้วยค่าที่วัดได้เป็นเลขฐาน 16 โดยแต่ละช่องจะถูกคั่นด้วย ',' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AI>0FD1,05A3,...,072E[CR]'

A	I	>	0	F	D	1	,	0	5	A	3	,	...	,	0	7	2	E	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	------

กรณีที่ต้องการอ่านค่า AI ทั้ง 24 ช่อง ให้ใช้ 'RAIXFFFFFF' แล้วจบด้วย '[CR]' ได้เลย

#	0	2	R	A	I	X	F	F	F	F	F	F	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	------

2. คำสั่งที่ใช้อ่านค่า Analog Input (Floating Point)

เริ่มต้นด้วย 'RAIFX' ตามด้วยช่องสัญญาณที่จะอ่าน โดยใช้รูปแบบของ bit ทั้งหมด 6 ไบต์ (MSB -> LSB, '0' = ไม่อ่าน, '1' = อ่าน) และจบด้วย '[CR]' เช่น อ่านค่า AI จากเครื่องหมายเลข 03 ช่องที่ 24, 23, 22, 18, 13, 10, 9, 5 จะได้คำสั่งดังนี้ '#03RAIFXE21310[CR]'

#	0	3	R	A	I	F	X	E	2	1	3	1	0	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	---	------

โดยตัวโมดูลจะตอบกลับมาเป็น 'AI>' ตามด้วยค่าที่วัดได้เป็นเลขทศนิยม ทำให้สามารถนำค่าที่ได้ไปใช้ต่อได้ทันที โดยแต่ละช่องจะถูกคั่นด้วย ',' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AI>12.1,470,...,-0.5[CR]'

A	I	>	1	2	.	1	,	4	7	0	,	...	,	-	0	.	5	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	------

กรณีที่ต้องการอ่านค่า AI ทั้ง 24 ช่อง ให้ใช้ 'RAIFXFFFFFF' แล้วจบด้วย '[CR]' ได้เลย

#	0	3	R	A	I	F	X	F	F	F	F	F	F	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	---	------

3. คำสั่งที่ใช้อ่านค่า Input/Output ทั้งหมด

ขึ้นต้นคำสั่งด้วย 'RADIOX' และจบด้วย '[CR]' ซึ่งจะได้คำสั่งดังนี้ '#09RADIOX[CR]'

#	0	9	R	A	D	I	O	X	[CR]
---	---	---	---	---	---	---	---	---	------

โดยตัวโมดูลจะตอบกลับมาเป็น 'AI>' ตามด้วยค่า AI เป็นเลขฐาน 16 ทั้ง 24 ช่อง โดยแต่ละช่องจะถูกคั่นด้วย ',' ตามด้วยค่า DI ทั้ง 4 ช่อง คั่นด้วย ',' ตามด้วยค่า DO ทั้ง 4 ช่อง และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AI>15.2,-9.83,...,0110,0011[CR]'

A	I	>	0	F	D	1	,	...	,	0	1	1	0	,	0	0	1	1	[CR]
---	---	---	---	---	---	---	---	-----	---	---	---	---	---	---	---	---	---	---	------

4. คำสั่งที่ใช้อ่านค่า Input/Output ทั้งหมด (Analog Floating Point).

ขึ้นต้นด้วย 'RADIOFX' และจบด้วย '[CR]' ซึ่งจะได้คำสั่งดังนี้ '#0ARADIOFX[CR]'

#	0	A	R	A	D	I	O	F	X	[CR]
---	---	---	---	---	---	---	---	---	---	------

โดยตัวโมดูลจะตอบกลับมาเป็น 'AI>' ตามด้วยค่า AI เป็นเลขทศนิยม ทั้ง 24 ช่อง โดยแต่ละช่องจะถูกคั่นด้วย ',' ตามด้วยค่า DI ทั้ง 4 ช่อง คั่นด้วย ',' ตามด้วยค่า DO ทั้ง 4 ช่อง และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AI>15.2,-9.83,...,0110,0011[CR]'

A	I	>	1	5	.	2	,	...	,	0	1	1	0	,	0	0	1	1	[CR]
---	---	---	---	---	---	---	---	-----	---	---	---	---	---	---	---	---	---	---	------

5. คำสั่งที่ใช้อ่านค่า R Shunt

ขึ้นต้นคำสั่งด้วย 'RRIX' ตามด้วยช่อง R Shunt ที่จะอ่าน โดยใช้รูปแบบของบิตทั้งหมด 6 บิต (MSB -> LSB, '0' = 'ไม่อ่าน', '1' = 'อ่าน') และจบด้วย '[CR]' เช่น อ่านค่า R Shunt จากเครื่องหมายเลข 0D ช่องที่ 23, 22, 17, 14, 10, 9, 8, 7, 6, 4, 3 จะได้คำสั่งดังนี้ '#0DRRIX6123EC[CR]'

#	0	D	R	R	I	X	6	1	2	3	E	C	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	------

โดยตัวโมดูลจะตอบกลับมาเป็น 'RIN>' ตามด้วยค่า R Shunt เป็นเลขทศนิยม โดยแต่ละช่องจะถูกคั่นด้วย ',' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'RIN>39.6,3.5,...,4.48[CR]'

R	I	N	>	3	9	.	6	,	3	.	5	,	...	,	4	.	4	8	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	------

กรณีที่ต้องการอ่านค่า R Shunt ทั้ง 24 ช่อง ให้ใช้ 'RRIXFFFFFF' แล้วจบด้วย '[CR]' ได้เลย

#	0	D	R	R	I	X	F	F	F	F	F	F	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	------

6. คำสั่งที่ใช้อ่านค่า Input Type (Analog Input – Expansion Module)

ขึ้นต้นคำสั่งด้วย 'RTYX' ตามด้วยช่องที่จะอ่านชนิดของ AI โดยใช้รูปแบบของ bit ทั้งหมด 6 บิต (MSB -> LSB, '0' = 'ไม่อ่าน', '1' = 'อ่าน') และจบด้วย '[CR]' เช่น อ่านค่า ชนิดของ AI จากเครื่องหมายเลข 0F ช่องที่ 23, 19, 17, 11, 7, 5, 3, 2, 1 จะได้คำสั่งดังนี้ '#0FRTYX450457[CR]'

#	0	F	R	T	Y	X	4	5	0	4	5	7	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	------

โดยตัวโมดูลจะตอบกลับมาในรูปแบบเดียวกับข้อ 15

T	Y	P	E	>	1	1	,	1	2	,	...	,	6	[CR]
---	---	---	---	---	---	---	---	---	---	---	-----	---	---	------

กรณีที่ต้องการอ่านชนิดของ AI ทั้ง 24 ช่อง ให้ใช้ 'RTYXFFFFFF' แล้วจบด้วย '[CR]' ได้เลย

#	0	F	R	T	Y	X	F	F	F	F	F	F	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	------

7. คำสั่งที่ใช้เขียนค่า R Shunt

ขึ้นต้นด้วย 'WRI' ตามด้วยช่อง R Shunt ที่จะเขียน คั่นด้วย '=' ตามด้วยค่าที่ต้องการ จะเขียนเป็นเลขทศนิยม และจบด้วย '[CR]' เช่น เขียนค่า R Shunt ให้กับเครื่องหมายเลข 13 ช่องที่ 5 = 245.75 จะได้คำสั่งดังนี้ '#13WRI5=247.5[CR]'

#	1	3	W	R	I	5	=	2	4	7	.	5	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	------

โดยตัวโมดูลจะตอบกลับมาเป็น 'RIN(' ตามด้วยช่อง R Shunt ที่จะเขียน ตามด้วย '>OK' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'RIN(5)>OK[CR]'

R	I	N	(5)	>	O	K	[CR]
---	---	---	---	---	---	---	---	---	------

ทั้งนี้ คำสั่งนี้ใช้เขียนค่าให้ R Shunt ได้ครั้งละ 1 ช่องเท่านั้น

8. คำสั่งที่ใช้กำหนดค่า Input Type (Analog Input)

ขึ้นต้นด้วย 'WTY' ตามด้วยชุดคำสั่งที่มี ช่องที่จะกำหนดชนิดของ AI ตามด้วย '=' ตามด้วยค่าที่ต้องการจะเขียนเป็นเลขฐาน 10 โดยแต่ละช่องจะคั่นด้วย ',' และจบด้วย '[CR]' เช่น กำหนดชนิดของ AI ให้กับเครื่องหมายเลข 14 ช่องที่ 1=1, 8=12, 21=9 จะได้คำสั่งดังตัวอย่างนี้ '#14WTY1=1,8=12,21=10[CR]'

#	1	4	W	T	Y	1	=	1	,	8	=	1	2	,	2	1	=	9	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------

โดยตัวโมดูลจะตอบกลับมาเป็น 'TYPE>OK' และจบด้วย '[CR]' ดังนี้

T	Y	P	E	>	O	K	[CR]
---	---	---	---	---	---	---	------

รหัสตอบกลับมาเมื่อเกิดข้อผิดพลาดในการส่งคำสั่งไปยังตัว EX24

ในกรณีที่การส่งคำสั่งไปยังตัวโมดูลนั้น หากชุดคำสั่งนั้นไม่ถูกต้อง ตัวโมดูลจะไม่ทำคำสั่งชุดนั้น และรายงานความผิดพลาดที่เกิดขึ้นกลับมาเป็นรหัสต่างๆ โดยจะขึ้นต้นด้วย 'ERR=' แล้วตามด้วยตัวเลข ตั้งแต่ 1-6 ดังนี้

- | | |
|----------------------------|---|
| 1 (illegal function) | คำสั่งไม่ถูกต้อง หรือโมดูลไม่รู้จักคำสั่งนี้ |
| 2 (illegal data address) | ค่าตำแหน่งเริ่มต้น เกินช่วงตำแหน่งที่กำหนดไว้ |
| 3 (illegal data value) | ค่าของข้อมูลที่ใช้ในชุดคำสั่งไม่ถูกต้อง
เช่น ค่าของ R Shunt ที่จะอ่าน ไม่ถูกต้อง |
| 4 (invalid data frame) | รูปแบบของชุดคำสั่งไม่ตรงตามข้อกำหนด
เช่น เขียนค่า Input Type โดยไม่มี ',' คั่นระหว่าง
หมายเลขช่องกับค่าที่จะเขียน |
| 6 (invalid number of byte) | จำนวนข้อมูลที่รับมาไม่ครบตามจำนวนที่แจ้งไว้ |

สรุปคำสั่งที่ใช้กับตัว EX24 (Wisco Protocol)

((H) = Heximal Value, (D) = Decimal Value, [CR] = carriage return)

Function	Command	EX24 Response
RAIX = Read AI Value (H)	#02RAIXA9C24F[CR]	AI>0FD1,05A3,...,072E[CR]
RAIFX = Read AI Value (D)	#03RAIFXE21310[CR]	AI>12.1,470,...,-0.5[CR]
RADIOX = Read All I/O	#09RADIOX[CR]	AI>0FD1,...,0110,0011[CR]
RADIOFX = Read All I/O (H)	#0ARADIOFX[CR]	AI>15.2,...,0110,0011[CR]
RRIX = Read R Shunt	#0DRRIX6123EC[CR]	RIN>39.6,3.5,...,4.48[CR]
RTYX = Read AI Type	#0FRTYX450457[CR]	TYPE>11,12,...,6[CR]
WRI = Write R Shunt	#13WRI5=247.5[CR]	RIN(5)>OK[CR]
WTY = Write AI Type	#14WTY1=1,8=12,21=9[CR]	TYPE>OK[CR]

การติดต่อกับโมดูลโดยใช้ **MODBUS(ASCII) Protocol**

EX24 สามารถใช้ Protocol MODBUS ในการติดต่อได้เช่นกัน โดยจะมีรูปแบบของคำสั่งดังต่อไปนี้ (CHAR = Character; 1 CHAR ประกอบไปด้วย 8 Data Bits, 1 Start Bit, และ 1 Stop Bit)

ADDR	FUNCTION	DATA	ERROR CHECK	EOF	READY TO REC RESP
2-CHAR 16-BITS	2-CHAR 16-BITS	N x 4-CHAR N x 16-BITS	2-CHAR 16-BITS	CR	LF

EX24 สนับสนุนฟังก์ชันพื้นฐานของ Modbus ทั้งหมด 1 ฟังก์ชัน ดังต่อไปนี้

MODBUS ASCII

READ INPUT REGISTERS (CODE 04)

Wisco

= Read Analog Input

การอ้าง Address บนตัวโมดูลมีดังนี้

Function Code	Reference	Address
04	Analog Input	3xxxx

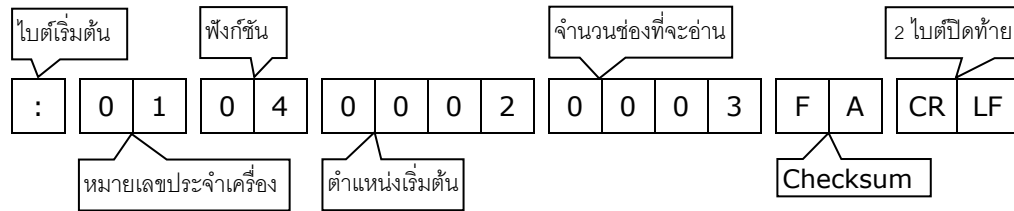
Analog Input Table

Name	Address
Analog Input Channel 9	30009
Analog Input Channel 10	30010
Analog Input Channel 11	30011
...	...
Analog Input Channel 23	30023
Analog Input Channel 24	30024

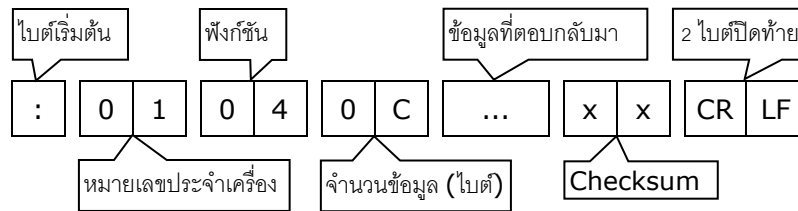
* รายละเอียดที่เหลือของ Modbus สามารถดูได้จาก 'Modbus Reference Guide' หรือที่ <http://www.modbus.org/specs.php>

ตัวอย่างฟังก์ชัน **MODBUS (ASCII) PROTOCOL**

Function Code 04



Response



วิธีการคิด CHECK SUM สำหรับ MODBUS(ASCII) Protocol

ใน **MODBUS Protocol** จะใช้ CHECK SUM ในการตรวจสอบความถูกต้องของข้อมูลที่ส่งไปทุกคำสั่ง การคิด CHECK SUM นั้นจะใช้การบวกข้อมูลทั้งหมดเข้าด้วยกัน (บวกเฉพาะข้อมูลที่เป็นตัวเลขเท่านั้น) บวกกันครั้งละ 1 ไบต์โดยค่าที่เกิน 1 byte นั้นเราจะตัดทิ้ง จากนั้น นำค่าที่ได้ 1 byte นั้น มาทำ 1's complement และ 2's complement เป็นอันเรียบร้อย

ตัวอย่างเช่น `: 0F 04 0001 0023 [CR] [LF]`

	HEXADECIMAL	BINARY
ไบต์เริ่มต้น	0FH	0000 1111
	04H	0000 0100
	00H	0000 0000
	01H	0000 0001
	00H	0000 0000
ไบต์สุดท้าย	23H	0010 0011
ผลลัพธ์	37H	0011 0111
คิดเฉพาะ 1 byte (8 bit)	37H	0011 0111
ทำ 1's complement (invert)	C8H	1100 1000
ทำ 2' complement	C8H + 1	1100 1000 + 1
ค่า Check sum ที่ได้	C9H	1100 1001

ข้อมูลที่จะส่งจึงเป็น `: 0F 04 0001 0023 C9 [CR] [LF]`

การแปลงข้อมูล Analog ชนิด Sign Integer

ข้อมูลชนิด sign integer ของ **EX24** นั้นได้จากการนำค่าวัดที่ได้ซึ่งเป็นค่าทศนิยม (IEEE741 Floating Point) มาทำการคูณเลื่อนจุดทศนิยม เพื่อให้ข้อมูลเป็นจำนวนเต็มสามารถเก็บในตัวแปร sign integer ขนาด 2 ไบต์ได้เพื่อลดขนาดข้อมูลลง ทำให้การส่งข้อมูลเร็วขึ้นและข้อมูลสามารถส่งผ่านข้อกำหนดของ MODBUS ได้ ดังนั้น ข้อมูลชนิด sign integer ที่ได้จึงต้องทำการเลื่อนจุดทศนิยมเข้าที่เดิมเสียก่อนโดยการหารด้วยค่าคงที่ตามตาราง ซึ่งค่าแต่ละชนิดจะไม่เท่ากันดังตารางข้างล่าง คอลัมน์สุดท้ายเป็นตัวหารสำหรับเลื่อนตำแหน่งทศนิยมเข้าที่เดิม

รหัส	ชนิดของสัญญาณ	ค่าวัด (Floating Point)	ค่าที่อ่านจากโมดูล	ตัวหารกลับ
00	Not Use	—	—	—
01	Thermocouple	R	0 – 1700 °C	0-1700
02		S	0 – 1700 °C	0-1700
03		K	(-)250.0 – 1300.0 °C	(-)2500-13000
04		E	0.0 – 1000.0 °C	0-10000
05		J	(-)200.0 – 700.0 °C	(-)2000-7000
06		T	(-)250.0 – 400.0 °C	(-)2500-4000
07		B	0 – 1800 °C	0-1800
08		R.T.D. Pt100	(-)200.0 – 800.0 °C	(-)2000-8000
09	Voltage(mV) 0 – 100	0.00 – 100.00 mV	0-10000	100
10	Voltage (V)	0 – 5	0.000 – 5.000 V	0-5000
11		0 – 10	0.000 – 10.000 V	0-10000
12	Current (mA)	0 – 20	0.00 – 20.00 mA	0-2000
13		0 – 40	0.00 – 40.00 mA	0-4000

Edit: 29/04/2022