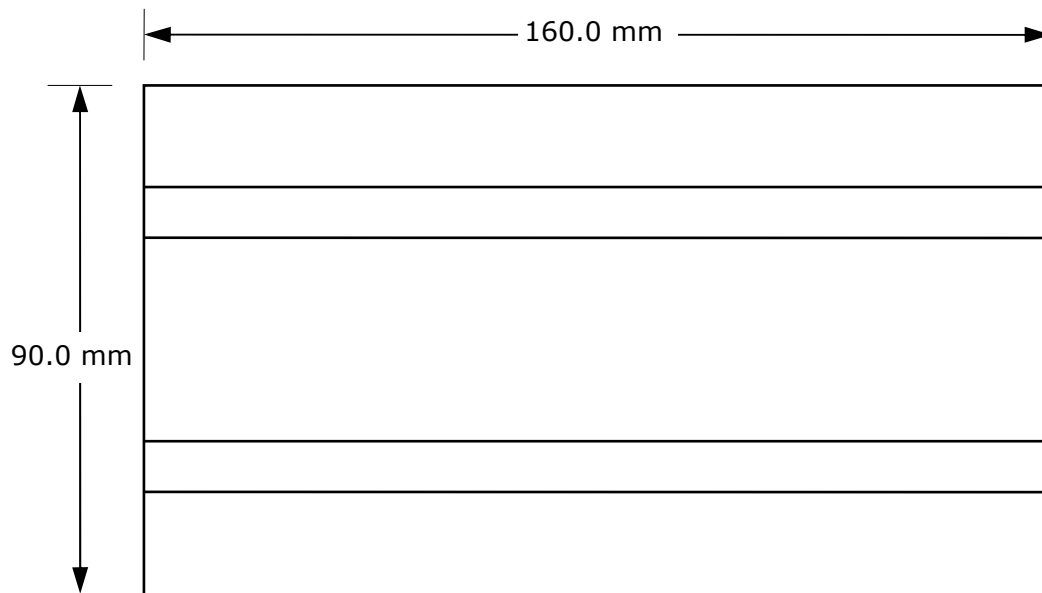


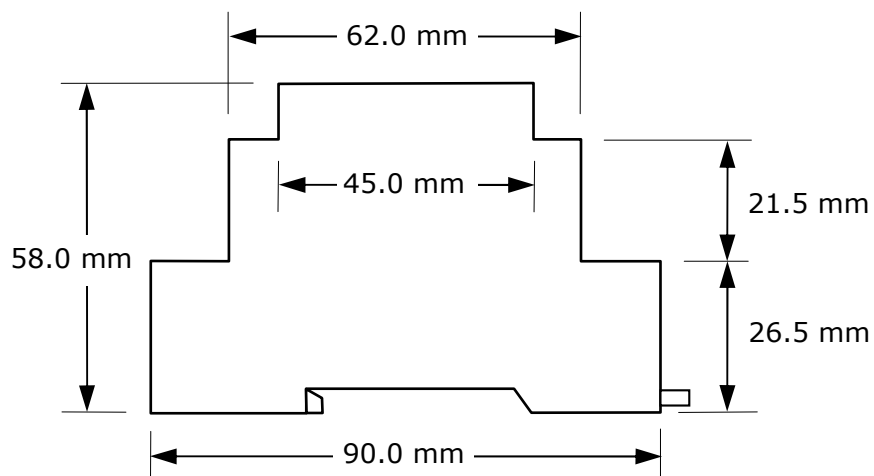
# Analog Output Module

## AO200

### ขนาดกล่อง (External Dimensions)



Top View



Side View

การเชื่อมต่อตัว **AO200** สามารถเชื่อมต่อได้สองมาตรฐานคือมาตรฐาน RS-232 และ RS-485 โดยมาตรฐาน RS-232 จะเป็นการเชื่อมต่อระหว่าง **AO200** กับ PC หนึ่งต่อหนึ่งเท่านั้น ส่วนมาตรฐาน RS-485 จะสามารถเชื่อมต่อกันได้ครั้งละหลายเครื่องโดยสามารถเชื่อมต่อ **AO200** ได้ทั้งหมด 32 เครื่อง พร้อมกันรวมกับ Computer อีก 1 เครื่อง โดยทั้งสองมาตรฐานจะใช้ข้อกำหนด (Protocol) เดียวกันในการติดต่อกับ **AO200** โดยมีรายละเอียดดังต่อไปนี้

### การติดต่อกับโมดูลโดยใช้ Wisco Protocol

ข้อมูลที่ใช้ในการติดต่อกับโมดูล **AO200** จะเป็นรหัส ASCII ทั้งหมดและในคำสั่งชุดหนึ่งจะประกอบไปด้วย



- ไบต์เริ่มต้น** ไบต์แรกที่บอกให้โมดูลรู้ว่าได้เริ่มต้นของชุดคำสั่ง โดยจะใช้อักขระ '#' เป็นตัวเริ่มต้น
- หมายเลขประจำเครื่อง** หมายเลขที่ใช้อ้างอิงตัวโมดูลสำหรับกรณีที่มีการต่อใช้งานพร้อมกันตั้งแต่ 2 ตัว ขึ้นไป โดยสามารถตั้งได้ที่ DIP Switch บนตัวโมดูล ซึ่งจะมีค่าตั้งแต่ 00h-1Fh และห้ามให้หมายเลขซ้ำกัน
- คำสั่ง** คำสั่งที่ใช้กับโมดูล สำหรับ **AO200** จะมีทั้งหมด 2 คำสั่ง
- ไบต์จบ** ไบต์สุดท้ายที่บอกให้โมดูลรู้ว่าสิ้นสุดของชุดคำสั่ง โดยจะใช้ [CR] (Carriage Return) ซึ่งเป็นอักขระตัวที่ 13 ในตาราง ASCII เป็นตัวปิดท้าย

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

ตัวอย่างการใช้งานคำสั่งสำหรับ **Wisco Protocol**

**รายละเอียดและตัวอย่างของคำสั่ง Wisco Protocol**

(    = 1 byte, CR = Carriage Return )

**1. คำสั่งที่ใช้อ่านค่า Analog Output**

เริ่มต้นด้วย 'RAO' ตามด้วยช่องสัญญาณที่จะอ่าน และจบด้วย '[CR]' เช่น อ่านค่า AI จากเครื่องหมายเลข 00 ช่องที่ 1, 2, 4 จะได้คำสั่งดังนี้ '#00RAO124[CR]'

#	0	0	R	A	O	1	2	4	CR
---	---	---	---	---	---	---	---	---	----

โดยตัวโมดูลจะตอบกลับมาเป็น 'AO>' ตามด้วยค่าที่วัดได้เป็นเลขฐาน16 โดยแต่ละช่องจะถูกคั่นด้วย ',' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AO>0FD1,05A3,072E[CR]'

A	O	>	0	F	D	1	,	0	5	A	3	,	0	7	2	E	CR
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

กรณีที่ต้องการอ่านค่า AO ทั้ง 8 ช่อง ให้ใช้ 'RAO' แล้วจบด้วย '[CR]' ได้เลย

#	0	0	R	A	O	CR
---	---	---	---	---	---	----

**2. คำสั่งที่ใช้เขียนค่า Analog Output**

เริ่มต้นด้วย 'WAO' ตามด้วยช่อง AO ที่จะเขียน ตามด้วย '=' ตามด้วยค่าที่ต้องการจะเขียนเป็นเลขฐาน16 และจบด้วย '[CR]' เช่น เขียนค่า AO ให้กับเครื่องหมายเลข 01 โดยให้ช่องที่ 4 = 01FC จะได้คำสั่งดังนี้ '#01WAO4=01FC[CR]'

#	0	1	W	A	O	4	=	0	1	F	C	CR
---	---	---	---	---	---	---	---	---	---	---	---	----

โดยตัวโมดูลจะตอบกลับมาเป็น 'WAO (' ตามด้วยช่อง AO ที่เขียน ตามด้วย ')>OK' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'WAO(4)>OK[CR]'

W	A	O	(	4	)	>	O	K	CR
---	---	---	---	---	---	---	---	---	----

ทั้งนี้ คำสั่งนี้ใช้เขียนค่าให้ AO ได้ครั้งละ 1 ช่องเท่านั้น

## รหัสตอบกลับมาเมื่อเกิดข้อผิดพลาดในการส่งคำสั่งไปยังตัวโมดูล AO200

ในกรณีที่การส่งคำสั่งไปยังตัวโมดูลนั้น หากชุดคำสั่งนั้นไม่ถูกต้อง ตัวโมดูลจะไม่ทำคำสั่งชุดนั้น และรายงานความผิดพลาดที่เกิดขึ้นกลับมาเป็นรหัสต่างๆ โดยจะขึ้นต้นด้วย 'ERR=' แล้วตามด้วยตัวเลข ตั้งแต่ 1-4 ดังนี้

1 (illegal function)	คำสั่งไม่ถูกต้อง หรือโมดูลไม่รู้จักคำสั่งนี้
2 (illegal data address)	ค่าตำแหน่งเริ่มต้น เกินช่วงตำแหน่งที่กำหนดไว้
3 (illegal data value)	ค่าของข้อมูลที่ใช้ในชุดคำสั่งไม่ถูกต้อง เช่น ค่าของ AO ที่จะอ่าน ไม่ถูกต้อง
4 (invalid data frame)	รูปแบบของชุดคำสั่งไม่ตรงตามข้อกำหนด เช่น เขียนค่า AO โดยไม่มี '=' คั่นระหว่างหมายเลขช่อง กับค่าที่จะเขียน

## สรุปคำสั่งที่ใช้กับตัวโมดูล AO200 (Wisco Protocol)

([CR] = carriage return)

<b>Function</b>	<b>Command</b>	<b>AO200 Response</b>
RAO = Read Analog Output	#00RAO124[CR]	AI>0FD1,05A3,072E[CR]
WAO = Write Analog Output	#01WAO04=01FC[CR]	WAO(4)>OK[CR]

## การติดต่อกับโมดูลโดยใช้ MODBUS(ASCII) Protocol

โมดูล **AO200** 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

โมดูล **AO200** สนับสนุนฟังก์ชันพื้นฐานของ Modbus ทั้งหมด 3 ฟังก์ชัน ดังต่อไปนี้

### **MODBUS ASCII**

READ OUTPUT REGISTERS (CODE 03)  
 PRESET SINGLE REGISTER (CODE 06)  
 PRESET MULTIPLE REGISTERS (CODE 16)

### **Wisco**

= Read Output  
 = Write Output  
 = Write Output

การอ้าง Address บนตัวโมดูลมีดังนี้

Function Code	Reference	Address
03, 06, 16	Analog Output	4xxxx

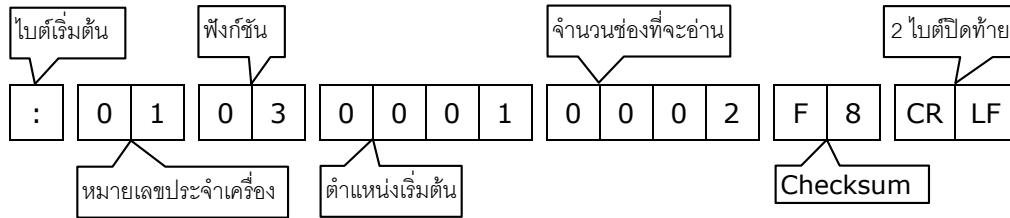
### **Analog Output Table**

Name	Address
Analog Output Channel 1	40001
Analog Output Channel 2	40002
Analog Output Channel 3	40003
Analog Output Channel 4	40004

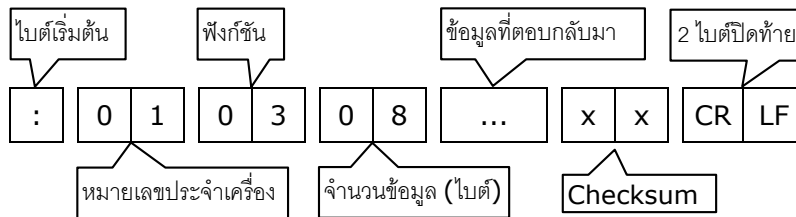
\*รายละเอียดที่เหลือของ Modbus สามารถดูได้จาก 'Modbus Reference Guide' หรือที่ <http://www.modbus.org/specs.php>

## ตัวอย่างฟังก์ชัน MODBUS(ASCII) PROTOCOL

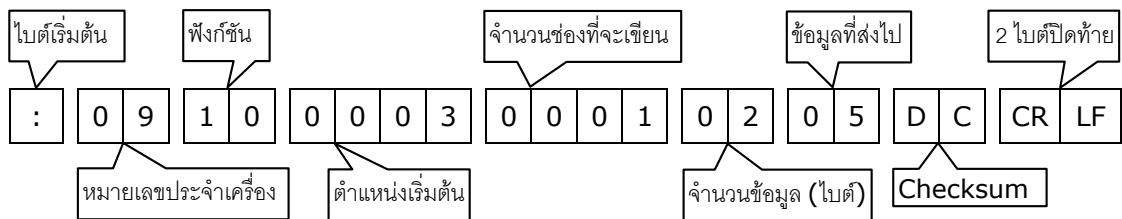
### Function Code 03



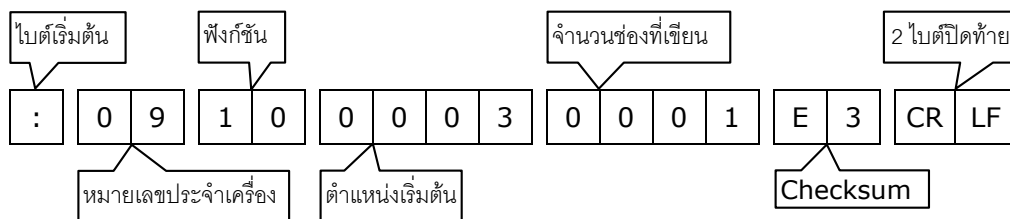
### Response



### Function Code 16



### 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 `: 0D 06 0002 03FF [CR] [LF]`

	HEXADECIMAL	BINARY
ไบต์เริ่มต้น	0DH	0000 1101
	06H	0000 0110
	00H	0000 0000
	02H	0000 0010
	03H	0000 0011
ไบต์สุดท้าย	FFH	1111 1111
ผลลัพธ์	117H	1 0001 0111
คิดเฉพาะ 1 byte (8 bit)	17H	0001 0111
ทำ 1's complement (invert)	E8H	1110 1000
ทำ 2' complement	E8H + 1	1110 1000 + 1
ค่า Check sum ที่ได้	E9H	1110 1001

ข้อมูลที่จะส่งจึงเป็น `: 0D 06 0002 03FF E9 [CR] [LF]`

### การตั้งค่าให้กับ **Dip Switch**

เมื่อแกะฝาด้านบนของโมดูลออก จะพบ Dipswitch ที่ใช้เลือก Station (ตำแหน่งที่ 1-5) และ Baud rate (ตำแหน่งที่ 6-7) ตามต้องการ และควรเลือกให้เหมาะสมกับการใช้งาน ซึ่งมีข้อควรพิจารณาดังนี้

- ความยาว และ ความต้านทานของสาย
  - การรบกวนจากภายนอก
  - ถ้าติดต่อผ่านโมเด็ม ไม่ควรตั้ง Baud rate สูงมากนัก ซึ่งจะขึ้นอยู่กับคุณภาพของคู่สายโทรศัพท์
- ส่วนการกำหนด Protocol ที่ใช้ติดต่อกับโมดูล ให้เลือก Dipswitch ตำแหน่งที่ 8 ดังนี้

'0' = MODBUS RTU, '1' = MODBUS ASCII / WISCO PROTOCOL.

### ตารางการตั้งค่า **Dip Switch**

1	2	3	4	5	Station
0	0	0	0	0	0 (00h)
1	0	0	0	0	1 (01h)
0	1	0	0	0	2 (02h)
1	1	0	0	0	3 (03h)
0	0	1	0	0	4 (04h)
1	0	1	0	0	5 (05h)
0	1	1	0	0	6 (06h)
1	1	1	0	0	7 (07h)
0	0	0	1	0	8 (08h)
1	0	0	1	0	9 (09h)
0	1	0	1	0	10 (0Ah)

1	2	3	4	5	Station
1	0	0	1	0	11 (0Bh)
0	0	1	1	0	12 (0Ch)
1	0	1	1	0	13 (0Dh)
0	1	1	1	0	14 (0Eh)
1	1	1	1	0	15 (0Fh)
0	0	0	0	1	16 (10h)
1	0	0	0	1	17 (11h)
0	1	0	0	1	18 (12h)
1	1	0	0	1	19 (13h)
0	0	1	0	1	20 (14h)
1	0	1	0	1	21 (15h)

1	2	3	4	5	Station
0	1	1	0	1	22 (16h)
1	1	1	0	1	23 (17h)
0	0	0	1	1	24 (18h)
1	0	0	1	1	25 (19h)
0	1	0	1	1	26 (1Ah)
1	1	0	1	1	27 (1Bh)
0	0	1	1	1	28 (1Ch)
1	0	1	1	1	29 (1Dh)
0	1	1	1	1	30 (1Eh)
1	1	1	1	1	31 (1Fh)

6	7	Baud rate
0	0	4800
1	0	9600
0	1	19200
1	1	57600

8	Protocol
0	MODBUS RTU
1	MODBUS ASCII / WISCO

แก้ไขครั้งสุดท้ายสุด 1 / มีนาคม / 2550