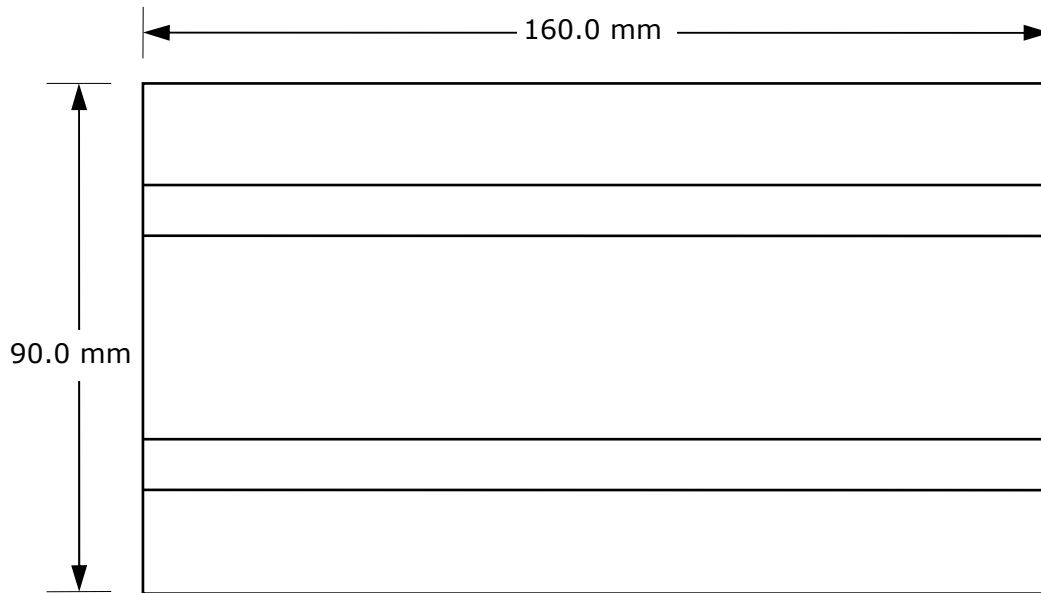


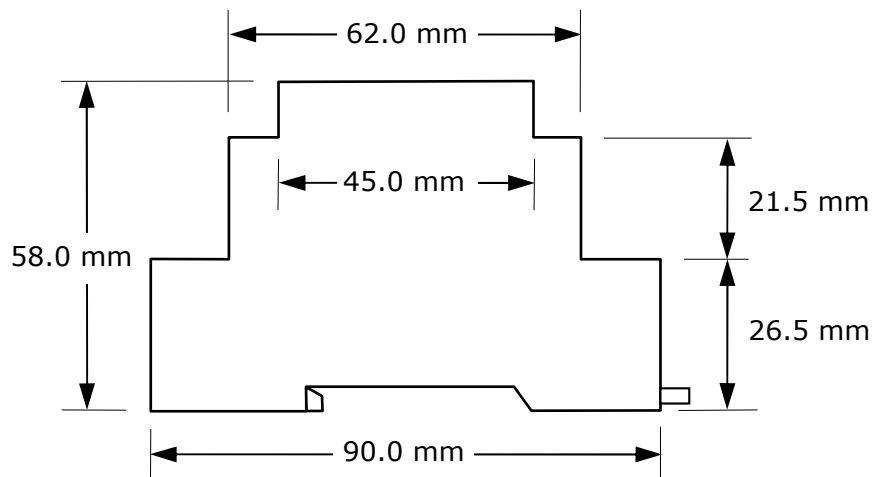
Analog Input Module

AI200

ขนาดกล่อง (External Dimensions)

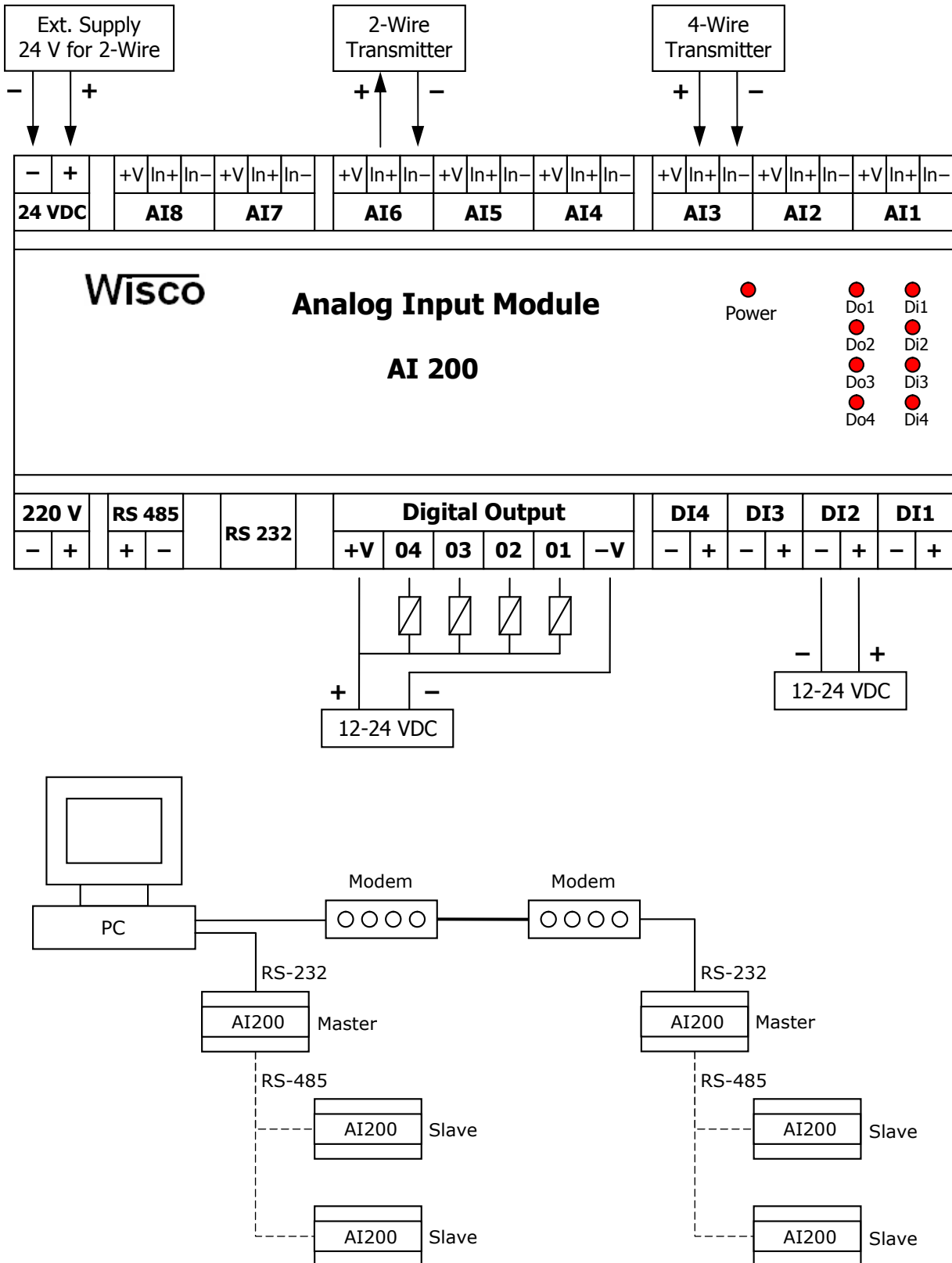


Top View



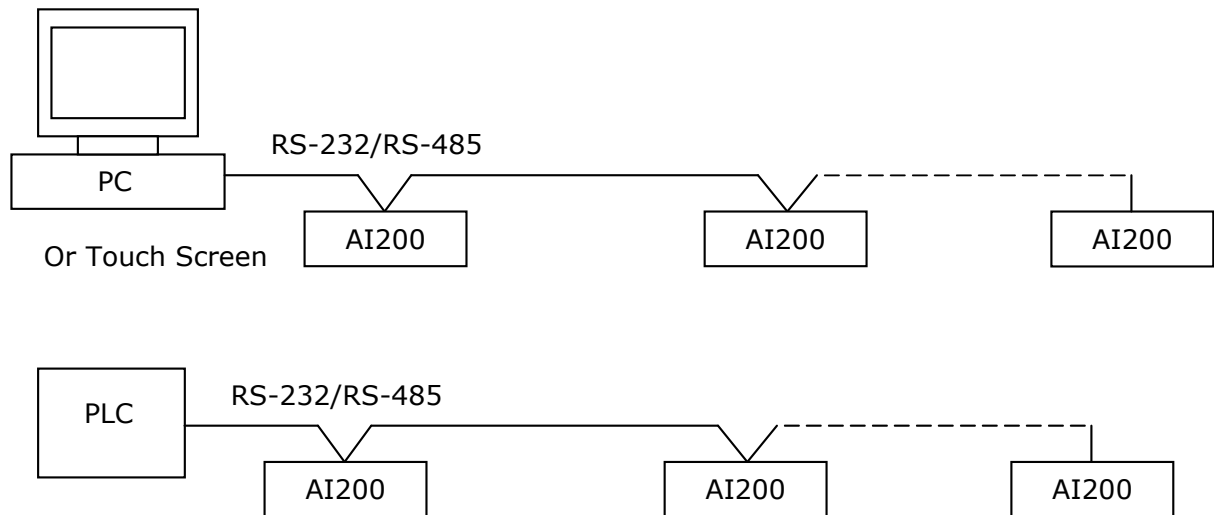
Side View

การต่อสาย (Wiring Diagram)



Specifications

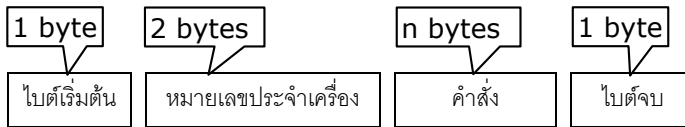
| | |
|---------------------|---|
| Analog Input | 8 Channels; 0-20 mA, 4-20 mA, 0-5 VDC, 1-5 VDC, 0-10 VDC |
| A/D Converter | 12 bits |
| Digital Input | 4 Channels, 24 VDC, Opto Isolated |
| Digital Output | 4 Channels, NPN Open Collector |
| Comm. Port | RS-232, RS-485 (Isolated) |
| Protocol | MODBUS (ASCII, RTU), ASCII Command |
| Software Support | Citect, Wonderware, LabView, Fix, Genesis, CIMPLICITY, DDE Server |
| Power Supply | 110 VAC, 220 VAC |
| Ambient Temperature | 0-50 °C |
| Mounting | DIN Rail |
| Dimension | W160 x H90 x D60 mm. |



การเชื่อมต่อตัว **AI200** สามารถเชื่อมต่อได้สองมาตรฐานคือมาตรฐาน RS-232 และ RS-485 โดยมาตรฐาน RS-232 จะเป็นการเชื่อมต่อระหว่าง **AI200** กับ PC หนึ่งต่อหนึ่งเท่านั้น ส่วนมาตรฐาน RS-485 จะสามารถเชื่อมต่อกันได้ครั้งละหลายเครื่องโดยสามารถเชื่อมต่อ **AI200** ได้ทั้งหมด 32 เครื่อง พร้อมกันรวมกับ Computer อีก 1 เครื่อง โดยทั้งสองมาตรฐานจะใช้ข้อกำหนด (Protocol) เดียวกันในการติดต่อกับ **AI200** โดยมีรายละเอียดดังต่อไปนี้

การติดต่อกับโมดูลโดยใช้ Wisco Protocol

ข้อมูลที่ใช้ในการติดต่อกับโมดูล **AI200** จะเป็นรหัส ASCII ทั้งหมดและในคำสั่งชุดหนึ่งจะประกอบไปด้วย



- ไบต์เริ่มต้น** ไบต์แรกที่บอกให้โมดูลรู้ว่าได้เริ่มต้นของชุดคำสั่ง โดยจะใช้อักขระ '#' เป็นตัวเริ่มต้น
- หมายเลขประจำเครื่อง** หมายเลขที่ใช้อ้างอิงตัวโมดูลสำหรับกรณีที่มีการต่อใช้งานพร้อมกันตั้งแต่ 2 ตัว ขึ้นไป โดยสามารถตั้งได้ที่ DIP Switch บนตัวโมดูล ซึ่งจะมีค่าตั้งแต่ 00h-1Fh และห้ามให้หมายเลขซ้ำกัน
- คำสั่ง** คำสั่งที่ใช้กับโมดูล สำหรับ **AI200** จะมีทั้งหมด 5 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

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

1. คำสั่งที่ใช้อ่านค่า Analog Input

เริ่มต้นด้วย 'RAI' ตามด้วยช่องสัญญาณที่จะอ่าน และจบด้วย '[CR]' เช่น อ่านค่า AI จากเครื่องหมายเลข 00 ช่องที่ 1, 2, 4, 5, 8 จะได้คำสั่งดังนี้ '#00RAI12458[CR]'

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|----|
| # | 0 | 0 | R | A | I | 1 | 2 | 4 | 5 | 8 | CR |
|---|---|---|---|---|---|---|---|---|---|---|----|

โดยตัวโมดูลจะตอบกลับมาเป็น 'AI>' ตามด้วยค่าที่วัดได้เป็นเลขฐาน16 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 8 ช่อง ให้ใช้ 'RAI' แล้วจบด้วย '[CR]' ได้เลย

| | | | | | | |
|---|---|---|---|---|---|----|
| # | 0 | 0 | R | A | I | CR |
|---|---|---|---|---|---|----|

ทั้งนี้ค่าที่ได้คือค่าที่มาจาก Analog-to-Digital Converter (ADC) 12 บิต (0-2047) จะต้องนำค่าไปคำนวณเทียบกับ Input ก่อนจึงจะใช้งานได้

2. คำสั่งที่ใช้อ่านค่า Digital Input

เริ่มต้นด้วย 'RDI' ตามด้วยช่องสัญญาณที่จะอ่าน และจบด้วย '[CR]' เช่น อ่านค่า DI จากเครื่องหมายเลข 04 ช่องที่ 2, 3, 4 จะได้คำสั่งดังนี้ '#04RDI234[CR]'

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|
| # | 0 | 4 | R | D | I | 2 | 3 | 4 | CR |
|---|---|---|---|---|---|---|---|---|----|

โดยตัวโมดูลจะตอบกลับมาเป็น 'DI>' ตามด้วยค่าที่วัดได้ ('0' = OFF, '1' = ON) ช่องละ 1 ไบต์ และจบด้วย '[CR]' ดังตัวอย่างนี้ 'DI>010[CR]'

| | | | | | | |
|---|---|---|---|---|---|----|
| D | I | > | 0 | 1 | 0 | CR |
|---|---|---|---|---|---|----|

กรณีที่ต้องการอ่านค่า DI ทั้ง 4 ช่อง ให้ใช้ 'RDI' แล้วจบด้วย '[CR]' ได้เลย

| | | | | | | |
|---|---|---|---|---|---|----|
| # | 0 | 4 | R | D | I | CR |
|---|---|---|---|---|---|----|

3. คำสั่งที่ใช้อ่านค่า Digital Output

คล้ายกับข้อ 2 แต่เปลี่ยนเป็นขึ้นต้นด้วย 'RDO' ตามด้วยช่องสัญญาณที่จะอ่าน และจบด้วย '[CR]' เช่น อ่านค่า DO จากเครื่องหมายเลข 05 ช่องที่ 1, 2, 4 จะได้คำสั่งดังนี้ '#05RDO124[CR]'

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|------|
| # | 0 | 5 | R | D | O | 1 | 2 | 4 | [CR] |
|---|---|---|---|---|---|---|---|---|------|

โดยตัวโมดูลจะตอบกลับมาเป็น 'DO>' ตามด้วยค่าที่วัดได้ ('0' = OFF, '1' = ON) ช่องละ 1 ไบต์ และจบด้วย '[CR]' ดังตัวอย่างนี้ 'DO>010[CR]'

| | | | | | | |
|---|---|---|---|---|---|------|
| D | O | > | 0 | 1 | 0 | [CR] |
|---|---|---|---|---|---|------|

กรณีที่ต้องการอ่านค่า DO ทั้ง 4 ช่อง ให้ใช้ 'RDO' แล้วจบด้วย '[CR]' ได้เลย

| | | | | | | |
|---|---|---|---|---|---|------|
| # | 0 | 5 | R | D | O | [CR] |
|---|---|---|---|---|---|------|

4. คำสั่งที่ใช้อ่านค่า Input/Output ทั้งหมด

เพื่อความสะดวกในการอ่านค่า AI, DI และ DO ทุกช่องพร้อมกัน ให้ใช้คำสั่งที่ขึ้นต้นด้วย 'RADIO' และจบด้วย '[CR]' ซึ่งจะได้คำสั่งดังนี้ '#07RADIO[CR]'

| | | | | | | | | |
|---|---|---|---|---|---|---|---|------|
| # | 0 | 7 | R | A | D | I | O | [CR] |
|---|---|---|---|---|---|---|---|------|

โดยตัวโมดูลจะตอบกลับมาเป็น 'AI>' ตามด้วยค่า AI เป็นเลขฐาน 16 ทั้ง 8 ช่อง โดยแต่ละช่องจะถูกคั่นด้วย ',' ตามด้วยค่า DI ทั้ง 4 ช่อง คั่นด้วย ',' ตามด้วยค่า DO ทั้ง 4 ช่อง และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AI>0FD1,05A3,...,0110,0011[CR]'

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

5. คำสั่งที่ใช้เขียนค่า Digital Output

ขึ้นต้นด้วย 'WDO' ตามด้วยช่องสัญญาณที่จะเขียน คั่นด้วย ',' ตามด้วยค่าที่ต้องการจะเขียนของช่องนั้น ('0' = OFF, '1' = ON) และจบด้วย '[CR]' เช่น เขียนค่า DO ไปที่เครื่องหมายเลข 11 ช่องที่ 1=OFF, 2=ON, 4=OFF จะได้คำสั่งดังนี้ '#11WDO124,010[CR]'

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|------|
| # | 1 | 1 | W | D | O | 1 | 2 | 4 | , | 0 | 1 | 0 | [CR] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|------|

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

| | | | | | |
|---|---|---|---|---|------|
| D | O | > | O | K | [CR] |
|---|---|---|---|---|------|

รหัสตอบกลับมาเมื่อเกิดข้อผิดพลาดในการส่งคำสั่งไปยังตัวโมดูล AI200

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 DO ที่จะอ่าน ไม่ถูกต้อง |
| 4 (invalid data frame) | รูปแบบของชุดคำสั่งไม่ตรงตามข้อกำหนด เช่น เขียนค่า DO โดยไม่มี ',' คั่นระหว่างหมายเลขช่องกับ ค่าที่จะเขียน |

สรุปคำสั่งที่ใช้กับตัวโมดูล AI200 (Wisco Protocol)

| Function | Command | AI200 Response |
|----------------------------|-----------------|---------------------------|
| RAI = Read Analog Input | #00RAI12458[CR] | AI>0FD1,05A3,...,072E[CR] |
| RDI = Read Digital Input | #04RDI234[CR] | DI>010[CR] |
| RDO = Read Digital Output | #05RDO[CR] | DO>1001[CR] |
| RADIO = Read All I/O | #07RADIO[CR] | AI>0FD1,...,0110,0011[CR] |
| WDO = Write Digital Output | #11WDO13,11[CR] | DO>OK[CR] |

การติดต่อกับโมดูลโดยใช้ MODBUS(ASCII) Protocol

โมดูล **AI200** 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 **AI200** สนับสนุนฟังก์ชันพื้นฐานของ Modbus ทั้งหมด 5 ฟังก์ชัน ดังต่อไปนี้

MODBUS ASCII

READ OUTPUT STATUS (CODE 01)
 READ INPUT STATUS (CODE 02)
 READ INPUT REGISTERS (CODE 04)
 FORCE SINGLE COIL (CODE 05)
 FORCE MULTIPLE COILS (CODE 15)

Wisco

= Read Digital Output
 = Read Digital Input
 = Read Analog Input
 = Write Digital Output
 = Write Digital Output

การอ้าง Address บนตัวโมดูลมีดังนี้

| Function Code | Reference | Address |
|---------------|----------------|---------|
| 01, 05, 15 | Digital Output | 0xxxx |
| 02 | Digital Input | 1xxxx |
| 04 | Analog Input | 3xxxx |

Digital Output Table

| Name | Address |
|--------------------------|---------|
| Digital Output Channel 1 | 00001 |
| Digital Output Channel 2 | 00002 |
| Digital Output Channel 3 | 00003 |
| Digital Output Channel 4 | 00004 |

Digital Input Table

| Name | Address |
|-------------------------|---------|
| Digital Input Channel 1 | 10001 |
| Digital Input Channel 2 | 10002 |
| Digital Input Channel 3 | 10003 |
| Digital Input Channel 4 | 10004 |

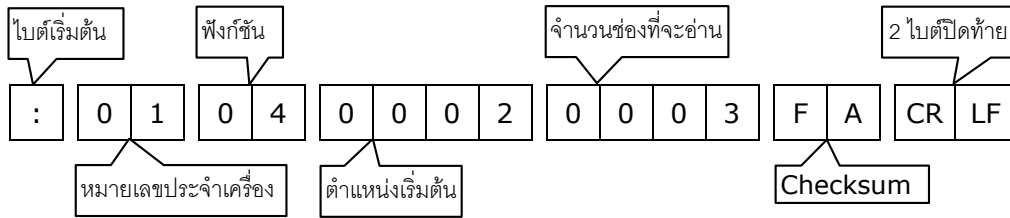
Analog Input Table

| Name | Address |
|------------------------|---------|
| Analog Input Channel 1 | 30001 |
| Analog Input Channel 2 | 30002 |
| Analog Input Channel 3 | 30003 |
| Analog Input Channel 4 | 30004 |
| Analog Input Channel 5 | 30005 |
| Analog Input Channel 6 | 30006 |
| Analog Input Channel 7 | 30007 |
| Analog Input Channel 8 | 30008 |

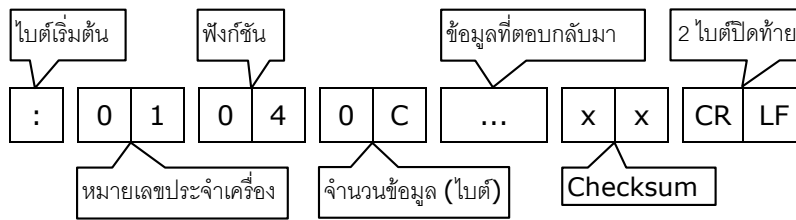
* รายละเอียดที่เหลือของ Modbus สามารถดูได้จาก 'Modbus Reference Guide' หรือที่ <http://www.modbus.org/specs.php>

ตัวอย่างฟังก์ชัน MODBUS(ASCII) PROTOCOL

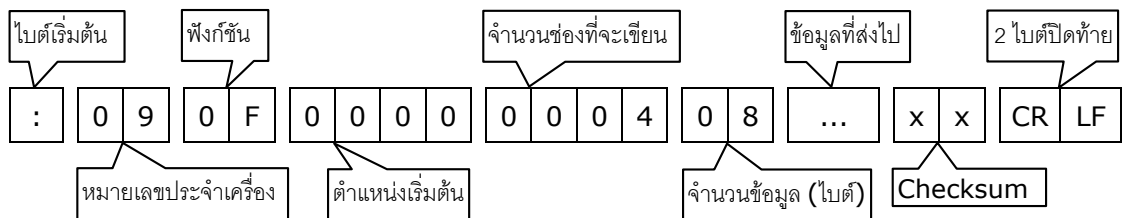
Function Code 04



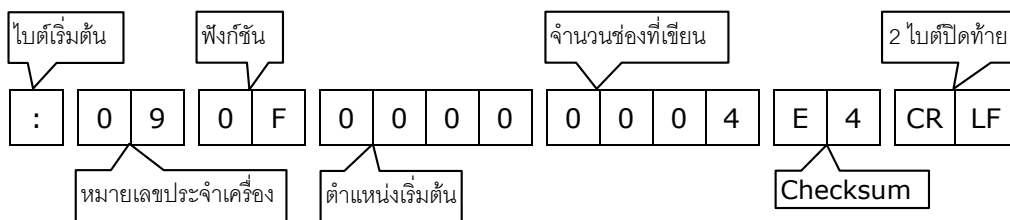
Response



Function Code 15



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]`

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