



Wisco AO200 Protocol



I. คุณสมบัติพิเศษ (Features)	1
II. การต่อสาย (Wiring Diagram)	2
1. การติดต่อกับโมดูลโดยใช้ Wisco ASCII Protocol	3
2. รายละเอียดและตัวอย่างของคำสั่ง Wisco Protocol	4
2.1 คำสั่งที่ใช้อ่านค่า Analog Output (Integer)	4
2.2 คำสั่งที่ใช้อ่านค่า Analog Output (Hex)	4
2.3 คำสั่งที่ใช้อ่านค่า Analog Output (Floating Point)	5
2.4 คำสั่งที่ใช้เขียนค่า Analog Output (Integer)	5
2.5 คำสั่งที่ใช้เขียนค่า Analog Output (Hex)	5
2.6 คำสั่งที่ใช้เขียนค่า Analog Output (Floating Point)	6
3. รหัสที่ตอบกลับมาเมื่อเกิดข้อผิดพลาดในการส่งคำสั่งไปยังโมดูล AO200	6
4. วิธีคิด CHECK SUM สำหรับ Wisco ASCII Protocol	7
5. การติดต่อกับโมดูลโดยใช้ MODBUS (ASCII) Protocol	8
6. วิธีคิด CHECK SUM สำหรับ MODBUS (ASCII) Protocol	10
7. ขนาดกล่อง (External Dimensions)	11

Analog Output Module

AO200

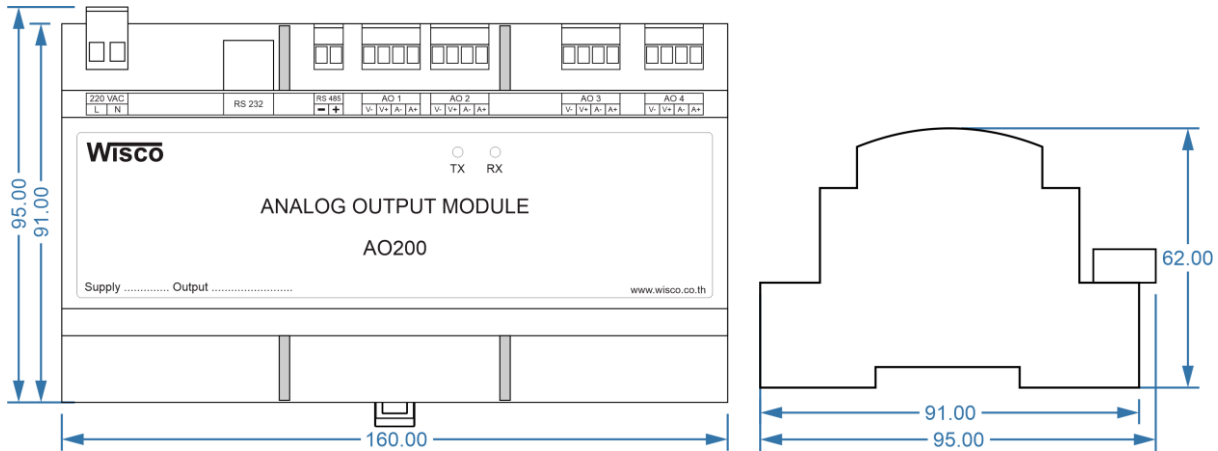


I. คุณสมบัติพิเศษ (Features)

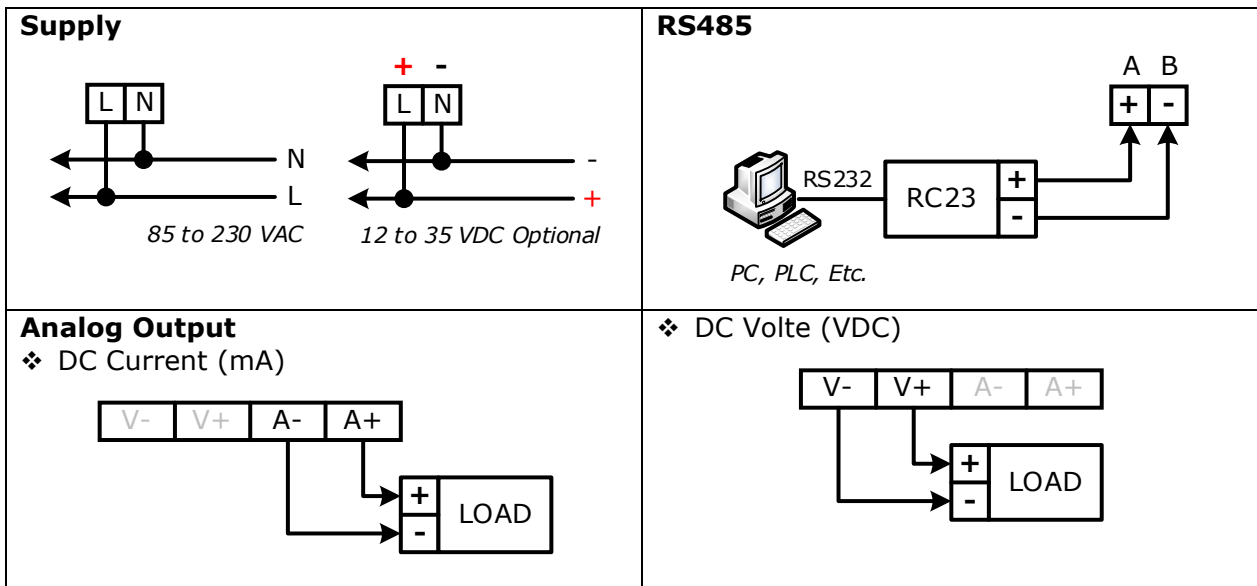
- ❖ 4 Channels Analog Output เลือกชนิดของสัญญาณได้ (Programmable Analog Output) 2 ชนิด ต่อ 1 channel
- ❖ แยกสัญญาณ (Isolation) ของ Analog Output ด้วย Opto Isolate
- ❖ การสื่อสารตามมาตรฐาน RS-232, RS-485 และ USB Port (Configuration)
- ❖ 3 ข้อกำหนดในการสื่อสาร (Communicate Protocol) ได้แก่ MODBUS RTU, MODBUS ASCII Protocol Compatible และ Wisco ASCII Protocol
- ❖ มีอัตราความเร็วในการ รับ/ส่ง ข้อมูลเลือกได้ 6 แบบ คือ 4800, 9600, 19200, 38400, 57600, 115200

II. การต่อสาย (Wiring Diagram)

Dimension (Unit: mm.)



Wiring



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

1. การติดต่อกับโมดูลโดยใช้ Wisco ASCII Protocol

ข้อมูลที่ใช้ในการติดต่อกับโมดูล AO200 จะเป็นรหัส ASCII ทั้งหมดและในคำสั่งชุดหนึ่งจะประกอบไปด้วย



ไบต์เริ่มต้น

ไบต์แรกที่บอกให้โมดูลรู้ว่าได้เริ่มต้นของชุดคำสั่ง โดยจะใช้อักขระ '#' เป็นตัวเริ่มต้น

หมายเลขประจำเครื่อง

หมายเลขประจำเครื่องที่ใช้อ้างอิงโมดูลสำหรับกรณีที่มีการต่อใช้งานพร้อมกันตั้งแต่ 2 เครื่องขึ้นไป และห้ามให้หมายเลขประจำเครื่องซ้ำกัน

คำสั่ง

คำสั่งที่ใช้กับโมดูล AO200 จะมีทั้งหมด 2 คำสั่ง

ไบต์จบ

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

Character	#	0	0	R	A	O	1	2	3	4	CR
ASCII Code	23H	30H	30H	52H	41H	4FH	31H	32H	33H	34H	0DH

ตัวอย่างการใช้งานคำสั่งสำหรับ Wisco ASCII Protocol

2. รายละเอียดและตัวอย่างของคำสั่ง Wisco Protocol

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

2.1 คำสั่งที่ใช้อ่านค่า Analog Output (Integer)

เริ่มต้นด้วย 'RAO' ตามด้วยช่องสัญญาณที่จะอ่าน และจบด้วย '[CR]' เช่น อ่านค่า AO จาก เครื่องหมายเลข 01 ช่องที่ 1, 2, 4 จะได้คำสั่งดังนี้ '#01RAOD124 [CR]'

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

โดยโมดูลจะตอบกลับมาเป็น 'AO>' ตามด้วยค่าที่อ่านได้เป็นเลขฐาน 10 โดยแต่ละช่องจะถูก คั่นด้วย ',' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AO>25, 50, 100 [CR]'

A	O	>	2	5	,	5	0	,	...	,	1	0	0	CR
---	---	---	---	---	---	---	---	---	-----	---	---	---	---	----

กรณีที่ต้องการอ่านค่า AO ทั้ง 4 ช่อง ให้ใช้คำสั่ง 'RAOD' แล้วจบด้วย '[CR]' ได้เลย

#	0	1	R	A	O	D	CR
---	---	---	---	---	---	---	----

2.2 คำสั่งที่ใช้อ่านค่า Analog Output (Hex)

เริ่มต้นด้วย 'RAOH' ตามด้วยช่องสัญญาณที่จะอ่าน และจบด้วย '[CR]' เช่น อ่านค่า AO จาก เครื่องหมายเลข 01 ช่องที่ 1, 2, 4 จะได้คำสั่งดังนี้ '#01RAOH124 [CR]' หรือใช้คำสั่ง '#01RAO124 [CR]'

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

หรือ

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

โดยโมดูลจะตอบกลับมาเป็น 'AO>' ตามด้วยค่าที่อ่านได้เป็นเลขฐาน 16 โดยแต่ละช่องจะถูก คั่นด้วย ',' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AO>0064, 0019, 0048 [CR]'

A	O	>	0	0	6	4	,	0	0	1	9	,	...	,	0	0	4	8	CR
---	---	---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	----

กรณีที่ต้องการอ่านค่า AO ทั้ง 4 ช่อง ให้ใช้คำสั่ง 'RAOH' หรือ 'RAO' แล้วจบด้วย '[CR]' ได้เลย

#	0	1	R	A	O	H	CR
---	---	---	---	---	---	---	----

หรือ

#	0	1	R	A	O	H	CR
---	---	---	---	---	---	---	----

2.3 คำสั่งที่ใช้อ่านค่า Analog Output (Floating Point)

เริ่มต้นด้วย 'RAOF' ตามด้วยช่องสัญญาณที่จะอ่าน และจบด้วย '[CR]' เช่น อ่านค่า AO จาก เครื่องหมายเลข 01 ช่องที่ 1, 2, 4 จะได้คำสั่งดังนี้ '#01RAOF124 [CR]'

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

โดยโมดูลจะตอบกลับมาเป็น 'AO>' ตามด้วยค่าที่อ่านได้เป็น Floating Point โดยแต่ละช่อง จะถูกคั่นด้วย ',' และจบด้วย '[CR]' ดังตัวอย่างนี้ 'AO>25.75, 10.12, 75.5 [CR]'

A	O	>	2	5	.	7	5	,	1	0	.	1	2	...	,	7	5	.	5	CR
---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	----

กรณีที่ต้องการอ่านค่า AO ทั้ง 4 ช่อง ให้ใช้คำสั่ง 'RAOF' แล้วจบด้วย '[CR]' ได้เลย

#	0	1	R	A	O	F	CR
---	---	---	---	---	---	---	----

2.4 คำสั่งที่ใช้เขียนค่า Analog Output (Integer)

เริ่มต้นด้วย 'WAO' ตามด้วยช่อง AO ที่จะเขียน ตามด้วย '=' ตามด้วยค่าที่ต้องการจะเขียน เป็นเลขฐาน 10 และจบด้วย '[CR]' เช่น เขียนค่า AO ให้กับเครื่องหมายเลข 01 โดยให้ ช่องที่ 4 = 25 จะได้คำสั่งดังนี้ '#01WAOD4=25 [CR]'

#	0	1	W	A	O	D	4	=	2	5	CR
---	---	---	---	---	---	---	---	---	---	---	----

โดยโมดูลจะตอบกลับมาเป็น 'WAO(4)>OK' จบด้วย '[CR]' ดังนี้ 'WAO(4)>OK [CR]'

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

*** คำสั่งนี้ใช้เขียนค่าให้ AO ได้ครั้งละ 1 ช่องเท่านั้น

2.5 คำสั่งที่ใช้เขียนค่า Analog Output (Hex)

เริ่มต้นด้วย 'WAOH' หรือ 'WAO' ตามด้วยช่อง AO ที่จะเขียน ตามด้วย '=' ตามด้วยค่าที่ต้องการจะเขียนเป็นเลขฐาน 16 และจบด้วย '[CR]' เช่น เขียนค่า AO ให้กับเครื่องหมายเลข 01 โดยให้ ช่องที่ 4 = 25 (ฐาน 10) จะได้คำสั่งดังนี้ '#01WAOH4=0019 [CR]' หรือใช้คำสั่ง '#01WAO4=0019 [CR]'

#	0	1	W	A	O	H	4	=	0	0	1	9	CR
---	---	---	---	---	---	---	---	---	---	---	---	---	----

หรือ

#	0	1	W	A	O	4	=	0	0	1	9	CR
---	---	---	---	---	---	---	---	---	---	---	---	----

โดยโมดูลจะตอบกลับมาเป็น 'WAO(4)>OK' จบด้วย '[CR]' ดังนี้ 'WAO(4)>OK [CR]'

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

*** คำสั่งนี้ใช้เขียนค่าให้ AO ได้ครั้งละ 1 ช่องเท่านั้น

2.6 คำสั่งที่ใช้เขียนค่า Analog Output (Floating Point)

ขึ้นต้นด้วย 'WAOF' ตามด้วยช่อง AO ที่จะเขียน ตามด้วย '=' ตามด้วยค่าที่ต้องการจะเขียน เป็น Floating Point และจบด้วย '[CR]' เช่น เขียนค่า AO ให้กับเครื่องหมายเลข 01 โดยให้ ช่องที่ 4 = 19.125 จะได้คำสั่งดังนี้ '#01WAOF4=19.125 [CR]'

#	0	1	W	A	O	F	4	=	1	9	.	1	2	5	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------

โดยโมดูลจะตอบกลับมาเป็น 'WAO(4)>OK' และจบด้วย '[CR]' ดังนี้ 'WAO(4)>OK [CR]'

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

*** คำสั่งนี้ใช้เขียนค่าให้ AO ได้ครั้งละ 1 ช่องเท่านั้น

3. รหัสที่ตอบกลับมาเมื่อเกิดข้อผิดพลาดในการส่งคำสั่งไปยังโมดูล AO200

ในกรณีที่ส่งคำสั่งไปยังโมดูลนั้น หากชุดคำสั่งนั้นไม่ถูกต้อง โมดูลจะไม่ทำคำสั่งชุดนั้น และรายงาน ความผิดพลาดที่เกิดขึ้นกลับมาเป็นรหัสต่างๆ โดยจะขึ้นต้นด้วย 'ERR=' แล้วตามด้วยตัวเลขตั้งแต่ 1-4 มีรายละเอียดดังนี้

- | | |
|--------------------------|---|
| 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 |

4. วิธีคิด CHECK SUM สำหรับ Wisco ASCII Protocol

AO200 จะใช้ CHECK SUM ในการตรวจสอบความถูกต้องของข้อมูลที่ส่งไปสำหรับ Read หรือ Write กับ EEPROM การคิด CHECK SUM นั้นจะใช้การบวกข้อมูลทั้งหมดเข้าด้วยกัน (บวกเฉพาะข้อมูลที่เป็นตัวเลขเท่านั้น) บวกกันครั้งละ 1 ไบต์โดยค่าที่เกิน 1 byte นั้นจะตัดทิ้ง จากนั้น นำค่าที่ได้ 1 ไบต์ นั้นมาทำ 2's complement

ตัวอย่างเช่น ``# 1A WEE 0 0000 05 11 22 33 44 55 [CR]'`

	HEXADECIMAL		BINARY
ไบต์เริ่มต้น	00H	}	0000 0000
	00H		0000 0000
	05H		0000 0000
	11H		0001 0001
	22H		0010 0010
	33H		0011 0011
	44H		0100 0100
ไบต์สุดท้าย	55H	}	0101 0101
ผลลัพธ์	104H		1 0000 0100
คิดเฉพาะ 1 byte (8 bit)	04H		0000 0100
ทำ 1's complement (invert)	FBH		1111 1011
ทำ 2' complement	FBH + 1		1111 1011+ 1
ค่า Check sum ที่ได้	FCH		1111 1100

ข้อมูลที่จะส่งจึงเป็น ``# 1A WEE 0 0000 05 11 22 33 44 55 FC [CR]'`

5. การติดต่อกับโมดูลโดยใช้ MODBUS (ASCII) Protocol

AO200 สามารถใช้ Protocol MODBUS ในการติดต่อได้เช่นกัน โดยจะมีรูปแบบของคำสั่งดังต่อไปนี้

ADDR	FUNCTION	DATA	ERROR CHECK
2-CHAR 16-BITS	2-CHAR 16-BITS	N x 4-CHAR N x 16-BITS	2-CHAR 16-BITS

AO200 สนับสนุนฟังก์ชันพื้นฐานของ Modbus ทั้งหมด 3 ฟังก์ชัน ดังต่อไปนี้

MODBUS ASCII

READ HOLDING REGISTER (CODE 03)
WRITE SINGLE REGISTER (CODE 06)
WRITE MULTIPLE REGISTERS (CODE 16)

Wisco

= Read Analog Output
= Write Analog Output
= Write Analog Output

การอ้าง Address ของโมดูลมีดังนี้

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

Analog Output Table (Integer)

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

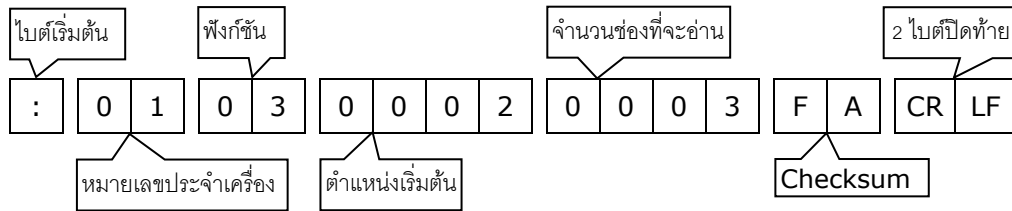
Analog Output Table (Floating Point)

Name	Address
Analog Output Channel 1	40101 - 40102
Analog Output Channel 2	40103 - 40104
Analog Output Channel 3	40105 - 40106
Analog Output Channel 4	40107 - 40108

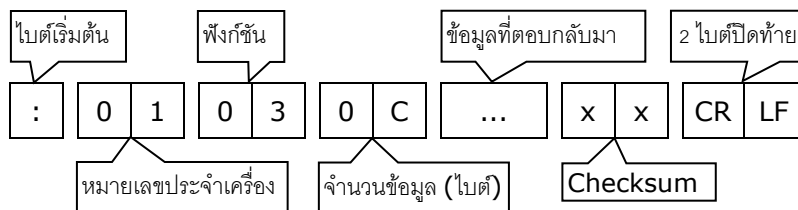
*** รายละเอียดที่เหลือของ Modbus สามารถดูได้จาก 'Modbus Reference Guide' หรือที่ <http://www.modbus.org/specs.php>

ตัวอย่างฟังก์ชัน MODBUS (ASCII) PROTOCOL

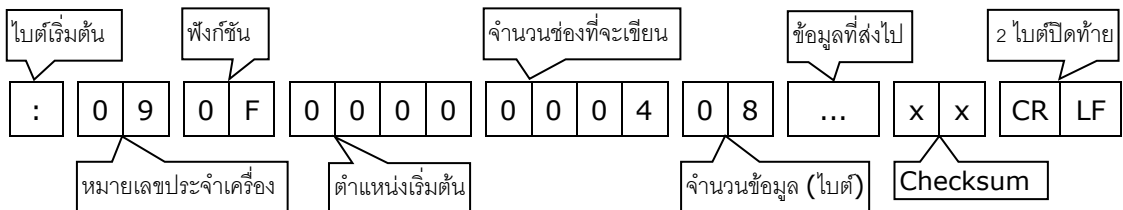
Function Code 03



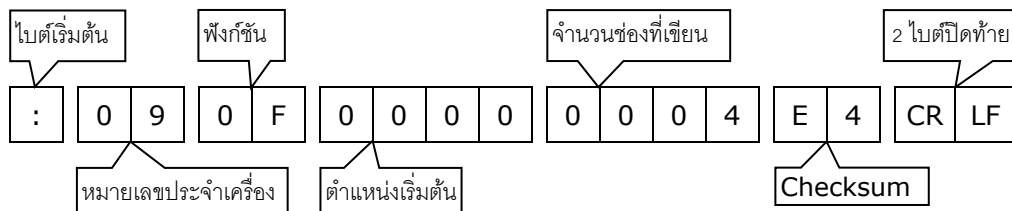
Response



Function Code 15



Response



6. วิธีคิด CHECK SUM สำหรับ MODBUS (ASCII) Protocol

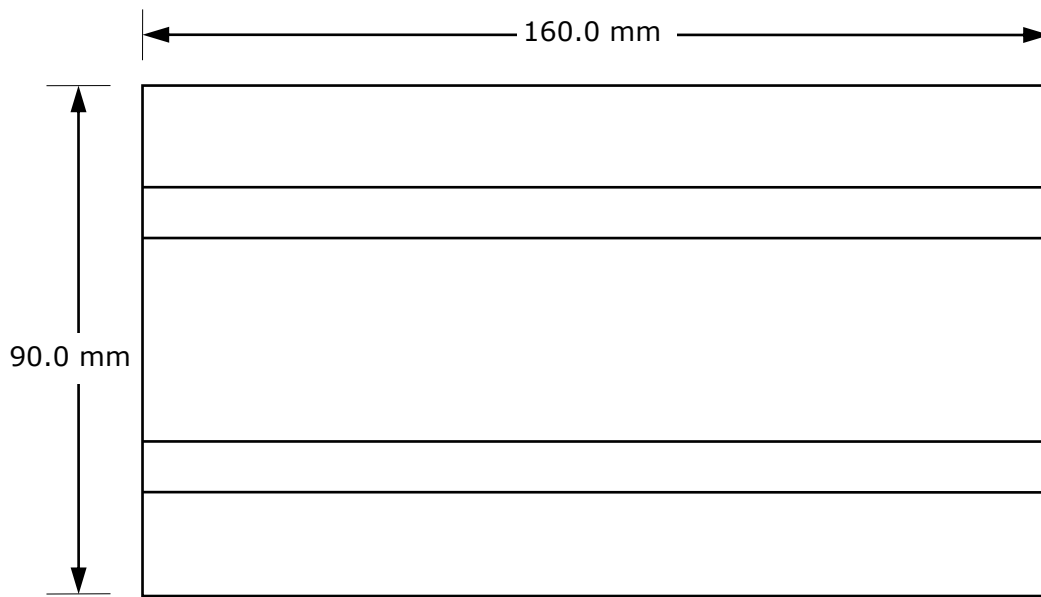
MODBUS Protocol จะใช้ CHECK SUM ในการตรวจสอบความถูกต้องของข้อมูลที่ส่งไปทุกคำสั่ง การคิด CHECK SUM นั้นจะใช้การบวกข้อมูลทั้งหมดเข้าด้วยกัน (บวกเฉพาะข้อมูลที่เป็นตัวเลขเท่านั้น) บวกกันครั้งละ 1 ไบต์โดยค่าที่เกิน 1 byte นั้นเราจะตัดทิ้ง จากนั้น นำค่าที่ได้ 1 byte นั้นมาทำ 2's complement

ตัวอย่างเช่น `: 0F 03 0001 0023 [CR] [LF]`

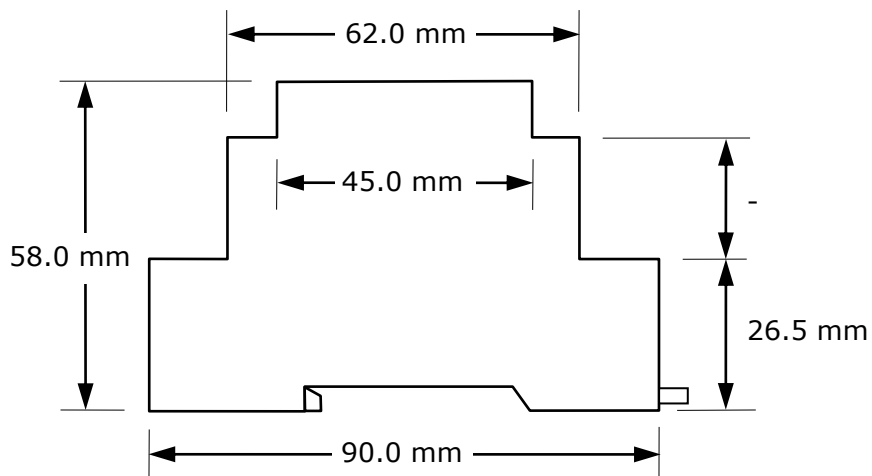
	HEXADECIMAL	BINARY
ไบต์เริ่มต้น	0FH	0000 1111
	03H	0000 0100
	00H	0000 0000
	01H	0000 0001
	00H	0000 0000
ไบต์สุดท้าย	23H	0010 0011
ผลลัพธ์	37H	0011 0111
คิดเฉพาะ 1 byte (8 bit)	37H	0011 0111
ทำ 1's complement (invert)	C8H	1100 1000
ทำ 2' complement	C8H + 1	1100 1000 + 1
ค่า Check sum ที่ได้	C9H	1100 1001

ข้อมูลที่จะส่งจึงเป็น `: 0F 03 0001 0023 C9 [CR] [LF]`

7. ขนาดกล่อง (External Dimensions)



Top View



Side View

Edit: 14/03/2017