



Wisco DL2200 Protocol



Data Logger DL2200 Protocol

การเชื่อมต่อ DL2200 สามารถเชื่อมต่อได้สองมาตรฐานคือมาตรฐาน RS-232 และ RS-485 โดยมาตรฐาน RS-232 จะเป็นการเชื่อมต่อระหว่าง DL2200 กับ PC หนึ่งต่อหนึ่งเท่านั้น ส่วนมาตรฐาน RS-485 จะสามารถเชื่อมต่อกันได้ครั้งละหลายเครื่องโดยสามารถเชื่อมต่อ DL2200 ได้ทั้งหมด 32 เครื่องพร้อมกันรวมกับ Computer อีก 1 เครื่อง โดยทั้งสองมาตรฐานจะใช้ข้อกำหนด (Protocol) เดียวกันในการติดต่อกับ DL2200 มีรายละเอียดดังนี้

การเชื่อมต่อกับโมดูลโดยใช้ Wisco ASCII Protocol

ข้อมูลที่ใช้ในการติดต่อกับโมดูล DL2200 จะเป็นรหัส ASCII ทั้งหมดและในคำสั่งชุดหนึ่งจะประกอบไปด้วย



ไบต์เริ่มต้น

ไบต์แรกที่บอกให้โมดูลรู้ว่าได้เริ่มต้นของชุดคำสั่ง โดยจะใช้อักขระ '#' เป็นตัวเริ่มต้น

หมายเลขประจำเครื่อง

หมายเลขประจำเครื่องที่ใช้อ้างอิงโมดูลสำหรับกรณีที่มีการต่อใช้งานพร้อมกันตั้งแต่ 2 เครื่องขึ้นไป โดยสามารถกำหนดได้ที่โปรแกรม Wisco Utility และที่ปุ่มกดหน้าเครื่อง ซึ่งจะมีค่าตั้งแต่ 1-254 และห้ามให้หมายเลขประจำเครื่องซ้ำกัน

คำสั่ง

คำสั่งที่ใช้กับโมดูล DL2200 จะมีทั้งหมด 5 คำสั่ง

ไบต์จบ

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

Character	#	0	0	R	A	I	CR
ASCII Code	23H	30H	30H	52H	41H	2AH	0DH

ตัวอย่างการใช้งานคำสั่งสำหรับ Wisco ASCII Protocol

รายละเอียดและตัวอย่างของคำสั่ง **Wisco ASCII Protocol**

(

--

 = 1 byte,

...

 = n bytes,

CR

 = Carriage Return)

1. คำสั่งที่ใช้อ่านค่า **Analog Input (Floating Point)**

เริ่มต้นด้วย 'RAI' (เป็นการค่า Analog Input ทั้งหมด รวมทั้งเชื่อมต่อกับ Module EX24) และจบด้วย '[CR]' เช่น อ่านค่า AI จากเครื่องหมายเลข "00" จะได้คำสั่งดังนี้ '#00RAI [CR]'

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

โดยโมดูลจะตอบกลับมาเป็น 'AI>' ตามด้วยค่าที่วัดได้เป็น Floating Point โดยแต่ละช่องจะถูกคั่นด้วย ',' และจบด้วย '[CR]' ตัวอย่างดังนี้ 'AI>50.58, 1.8, ..., 11.8 [CR]'

A	I	>	5	0	.	5	8	,	1	.	8	,	...	,	1	1	.	8	CR
---	---	---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	----

ทั้งนี้ค่าที่ได้ต้องนำมาแปลงก่อนจึงจะได้ค่าที่ถูกต้อง โดยสามารถดูรายละเอียดได้ที่ ตารางการแปลงข้อมูล Analog ชนิด Sign Integer ที่อยู่บนหน้าสุดท้าย

2. คำสั่งที่ใช้อ่านค่า **Digital Input**

เริ่มต้นด้วย 'RDI' และจบด้วย '[CR]' ซึ่งจะทำการอ่านค่า DI ทั้ง 4 ช่อง ดังนี้

#	0	0	R	D	I	CR
---	---	---	---	---	---	----

โดยโมดูลจะตอบกลับมาเป็น 'DI>' ตามด้วยค่าที่วัดได้ ('0' = OFF, '1' = ON) ช่องละ 1 ไบต์ และจบด้วย '[CR]' ตัวอย่างดังนี้ 'DI>0010[CR]'

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

3. คำสั่งที่ใช้อ่านค่า **Digital Output**

เริ่มต้นด้วย 'RDO' และจบด้วย '[CR]' ซึ่งจะทำการอ่านค่า DO ทั้ง 4 ช่อง ดังนี้

#	0	0	R	D	O	CR
---	---	---	---	---	---	----

โดยโมดูลจะตอบกลับมาเป็น 'DO>' ตามด้วยค่าที่วัดได้ ('0' = OFF, '1' = ON) ช่องละ 1 ไบต์ และจบด้วย '[CR]' ตัวอย่างดังนี้ 'DO>0101[CR]'

D	O	>	0	1	0	1	CR
---	---	---	---	---	---	---	----

4. คำสั่งที่ใช้อ่านค่า **Counter (DI 4)**

ขึ้นต้นด้วย 'RCT' และจบด้วย '[CR]' ดังนี้

#	0	0	R	C	T	[CR]
---	---	---	---	---	---	------

โดยโมดูลจะตอบกลับมาเป็น 'CT>' ตามด้วยค่าที่วัดได้ และจบด้วย '[CR]' ตัวอย่างดังนี้
'CT>15.8 [CR]'

C	T	>	1	5	.	8	[CR]
---	---	---	---	---	---	---	------

5. คำสั่งที่ใช้อ่านค่า **Input/Output ทั้งหมด**

สามารถอ่านค่า AI, DI, DO และ CT ทุกช่องพร้อมกัน โดยใช้คำสั่งที่ขึ้นต้นด้วย 'RAL' และจบด้วย '[CR]' ซึ่งจะได้คำสั่งดังนี้ '#00RAL [CR]'

#	0	0	R	A	L	[CR]
---	---	---	---	---	---	------

โดยโมดูลจะตอบกลับมาเป็น 'AI>' ตามด้วยค่า AI (Floating Point) ทั้ง 24 ช่อง โดยแต่ละช่องจะถูกคั่นด้วย ',' และปิดท้ายด้วย ';' ตามด้วยค่า DI ทั้ง 4 ช่อง คั่นด้วย ',' และปิดท้ายด้วย ';' ตามด้วยค่า DO ทั้ง 4 ช่อง คั่นด้วย ',' และปิดท้ายด้วย ';' และตามด้วยค่า CT คั่นด้วย ',' และปิดท้ายด้วย ';' จบด้วย '[CR]' ตัวอย่างดังนี้ 'ALL> AI, 21.57, ..., 8.21; DI, 1, 0, 0, 1; DO, 0, 1, 1, 1; CT, 15.57; [CR]'

A	L	L	>	A	I	,	2	1	.	5	7	,	...	,	8	.	2	1	;
D	I	,	1	,	0	,	0	,	1	;									
D	O	,	0	,	1	,	1	,	1	;									
C	T	,	1	5	.	5	7	;											

6. คำสั่งที่ใช้เขียนค่า **Digital Output**

ขึ้นต้นด้วย 'WDO=' ตามด้วยค่าที่ต้องการจะเขียนของช่องที่ต้องการ ('0' = OFF, '1' = ON) โดยแต่ละช่องคั่นด้วย ',' และจบด้วย '[CR]' เช่น เขียนค่า DO ไปที่เครื่อง หมายเลข 11 ช่องที่ 1=OFF, 2=ON, 3=ON, 4=OFF จะได้คำสั่งดังนี้ '#11WDO= 0,1,1,0 [CR]'

#	1	1	W	D	O	=	0	,	1	,	1	,	0	[CR]
---	---	---	---	---	---	---	---	---	---	---	---	---	---	------

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

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

รหัสที่ตอบกลับมาเมื่อเกิดข้อผิดพลาดในการส่งคำสั่งไปยังโมดูล **DL2200**

ในกรณีที่ส่งคำสั่งไปยังโมดูลนั้น หากชุดคำสั่งนั้นไม่ถูกต้อง โมดูลจะไม่ทำคำสั่งชุดนั้น และรายงานความผิดพลาดที่เกิดขึ้นกลับมาเป็นรหัสต่างๆ โดยจะขึ้นต้นด้วย 'ERR=' แล้วตามด้วยตัวเลขตั้งแต่ 1-6 มีรายละเอียดดังนี้

- | | |
|----------------------------|---|
| 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 ',' คั่นระหว่างหมายเลขช่องกับค่าที่จะเขียน |
| 5 (check sum error) | ค่า check sum ไม่ถูกต้อง (อาจเกิดจากความผิดพลาดระหว่างส่งข้อมูล) |
| 6 (invalid number of byte) | จำนวนข้อมูลที่ได้รับมาไม่ครบตามจำนวนที่แจ้งไว้ |

การเชื่อมต่อกับโมดูลโดยใช้ **MODBUS (ASCII) Protocol**

DL2200 สามารถใช้ Protocol MODBUS ในการเชื่อมต่อได้เช่นกัน 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

DL2200 สนับสนุนฟังก์ชันพื้นฐานของ Modbus ทั้งหมด 8 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

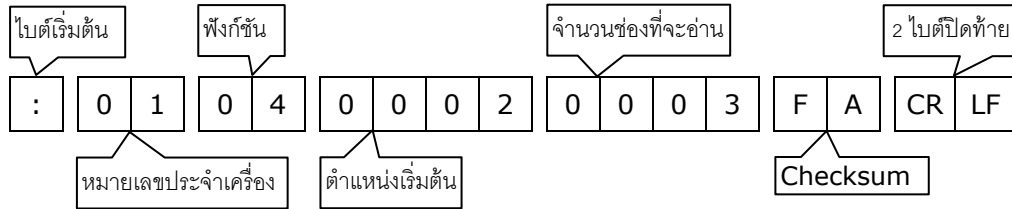
Analog Input Table (Floating Point)

Name	Address
Analog Input Channel 1	30001-30002
Analog Input Channel 2	30003-30004
Analog Input Channel 3	30005-30006
Analog Input Channel 4	30007-30008
Analog Input Channel 5	30009-30010
Analog Input Channel 6	30011-30012
Analog Input Channel 7	30013-30014
Analog Input Channel 8	30015-30016
⋮	⋮
⋮	⋮
⋮	⋮
Analog Input Channel 24	30047-30048

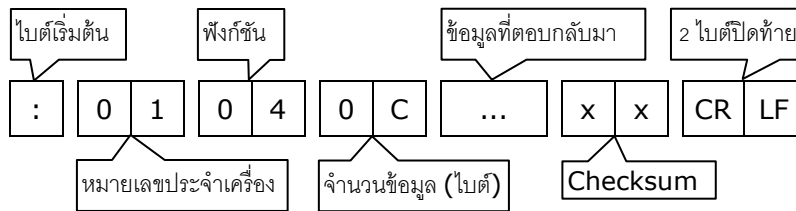
*** รายละเอียดที่เหลือของ Modbus สามารถดูได้จาก 'Modbus Reference Guide' หรือที่ <http://www.modbus.org/specs.php>

ตัวอย่างฟังก์ชัน MODBUS (ASCII) PROTOCOL

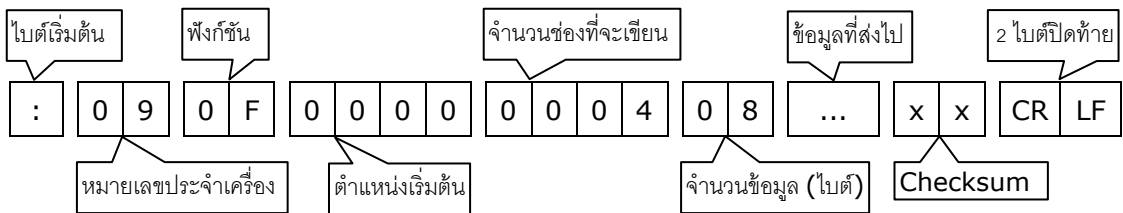
Function Code 04



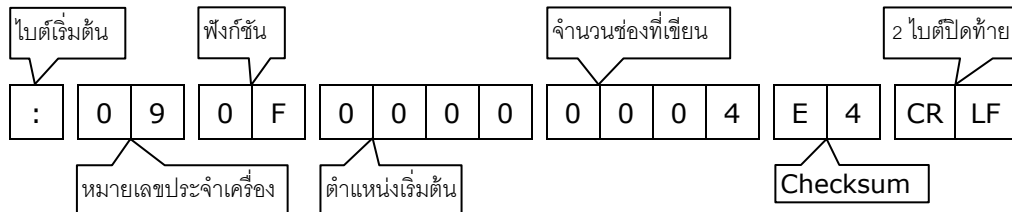
Response



Function Code 15



Response



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 1's complement และ 2's complement

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]`

Edit: 14/02/2019