

ET-Modbus RTU SHT31 BOX V2

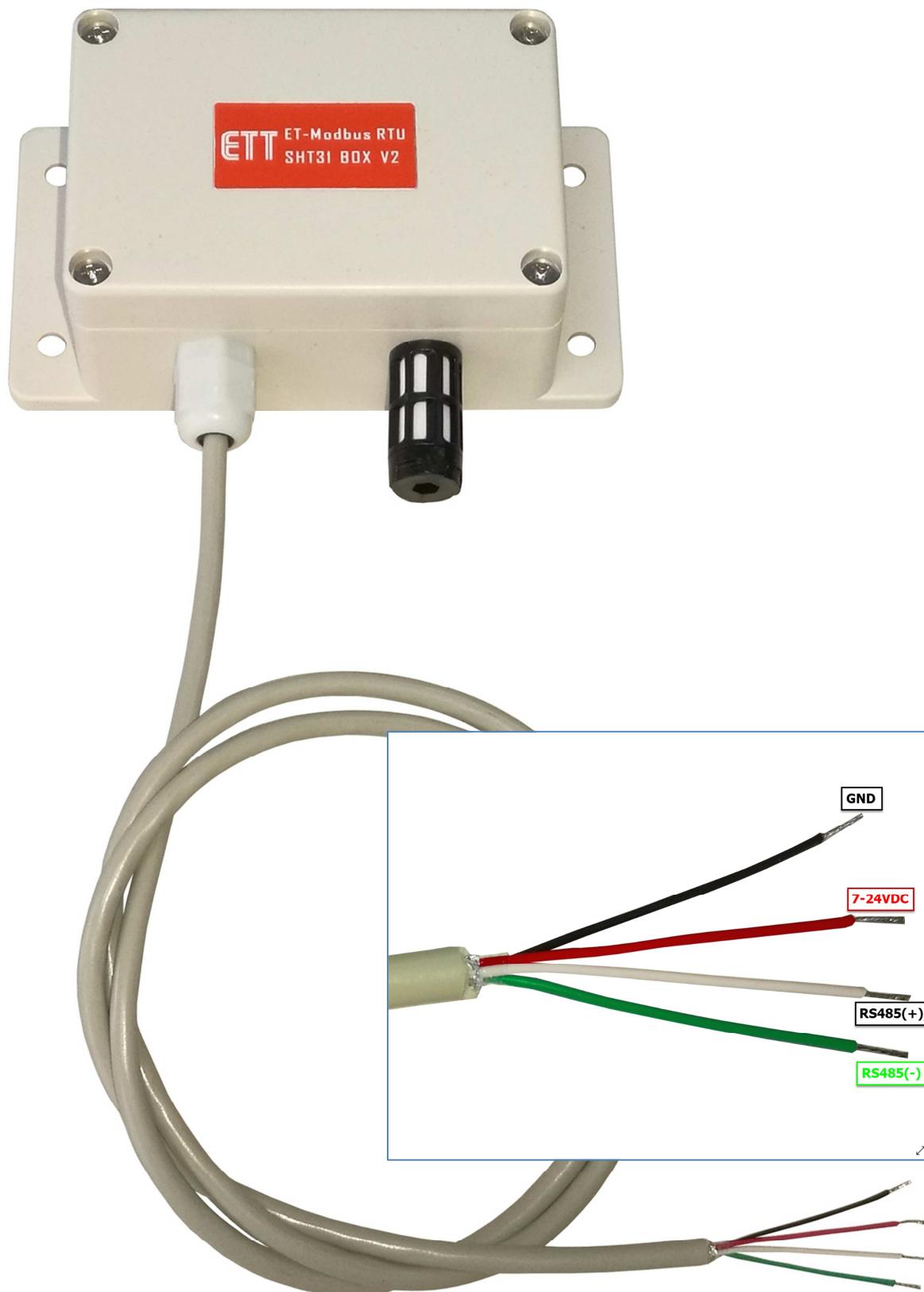


ET-Modbus RTU SHT31 BOX V2 เป็นชุดวัดอุณหภูมิและความชื้น โดยใช้เซ็นเซอร์ SHT31 โดยสามารถต่อสื่อสารด้วย Modbus RTU Protocol หรือ ASCII Command ผ่านทางการสื่อสาร RS485 แบบ Half Duplex สามารถเชื่อมต่อสั่งงานและอ่านค่าผ่านระบบเครือข่าย RS485 มาตรฐาน บรรจุในกล่องพลาสติก ABS กันน้ำ เหมาะสำหรับติดตั้งใช้งานในโรงเรือนและกลางแจ้งได้เป็นอย่างดี

คุณสมบัติของ ET-Modbus RTU SHT31 BOX V2

- ใช้กับแหล่งจ่ายไฟ 7-24VDC
- สื่อสารด้วย RS485 Half Duplex สามารถกำหนดหมายเลข Slave Device ID ได้อิสระจากคำสั่ง
- การเขียนต่อสื่อสารผ่านสัญญาณแบบ RS485 Half Duplex ด้วย Protocol แบบ Modbus RTU หรือ ASCII Command โดยสามารถเลือกได้จาก Jumper(RTU/ASCII) ตามต้องการ
 - สามารถกำหนดค่า Slave Address ได้อิสระ(1-247)จาก คำสั่ง
 - ใช้การสื่อสารแบบ RS485 Half duplex
 - Baudrate 9600bps
 - Data 8Bit
 - None Parity
 - 1 Stop Bit
- มีคำสั่งรองรับการทำงานแบบ
 - Read Input Register สำหรับอ่านค่า
 - Firmware Version
 - Temperature Sensor
 - Humidity Sensor
 - Write Single Holding Register สำหรับกำหนดค่าต่างๆ
 - Slave ID
 - Temperature Adjust(ชดเชยความผิดพลาดของอุณหภูมิ)
 - Humidity Adjust(ชดเชยความผิดพลาดของความชื้น)
 - Write Multiple Holding Register สำหรับกำหนดค่า ต่างๆ
 - Slave ID
 - Temperature Adjust(ชดเชยความผิดพลาดของอุณหภูมิ)
 - Humidity Adjust(ชดเชยความผิดพลาดของความชื้น)
- 1 Sensor SHT31 บรรจุใน COVER แบบกันน้ำ
 - ย่านการวัดอุณหภูมิ -40°C ถึง +125°C
 - ค่าความแม่นยำ +/- 0.3 °C
 - ย่านการวัดความชื้น 0 %RH ถึง 100 %RH
 - ค่าความแม่นยำ +/- 2.0 %RH
- ใช้ขัวต่อสัญญาณแบบสายต่อร้อยผ่านขัวต่อ กันน้ำ Cable Grand
- บรรจุในกล่องพลาสติก ABS กันน้ำ ขนาด
 - 100mm. x 68mm x 50mm. (L X W X H)
 - 3.93" x 2.67" x 1.96" (L X W X H)





รูปแสดงสายสัญญาณของบอร์ด

Modbus Register ของ ET-Modbus RTU SHT31 BOX V2

ET-Modbus RTU SHT31 BOX V2 จะใช้ Modbus Register จำนวน 2 กลุ่ม สำหรับเป็นตัวกลางในการสื่อสารระหว่างบอร์ดกับอุปกรณ์ภายนอกผ่าน Modbus RTU Protocol ดังนี้

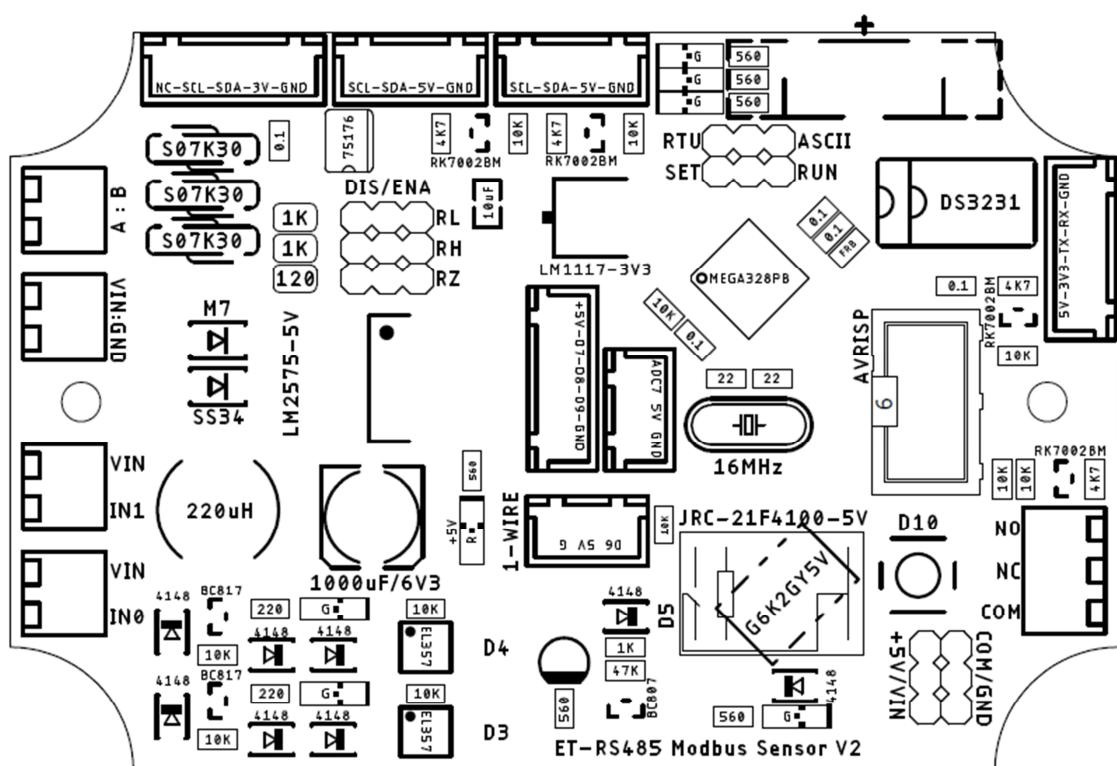
Holding Register (16 Bit Read/Write Register)		
Address	Function	Data Type
0000	Read/Write: Configuration Slave ID(1..247)	Unsigned 16 Bit (1..247)
0001	Read/Write: Temperature Adjust(Signed/10)	Signed 16 Bit
0002	Read/Write: Humidity Adjust(Signed/10)	Signed 16 Bit
0003	Read/Write: Temperature Adjust(Float)	Float 32Bit (Big-endian)
0004		
0005	Read/Write: Humidity Adjust(Float)	Float 32Bit (Big-endian)
0006		

- Holding Register[0] ใช้กำหนดค่าหมายเลข Slave ID ของบอร์ด กำหนดค่าได้ระหว่าง 1-247
- Holding Register[1] ใช้กำหนดค่าชดเชยความผิดพลาดของอุณหภูมิ เป็นค่าแบบเลขจำนวนเต็ม Signed/10 เช่นถ้ากำหนดเป็น 1 จะมีค่าจริงๆ เป็น 1/10 หรือ 0.1
- Holding Register[2] ใช้กำหนดค่าชดเชยความผิดพลาดของความชื้น เป็นค่าแบบเลขจำนวนเต็ม Signed/10 เช่นถ้ากำหนดเป็น 1 จะมีค่าจริงๆ เป็น 1/10 หรือ 0.1
- Holding Register[3:4] ใช้กำหนดค่าชดเชยความผิดพลาดของอุณหภูมิ เป็นค่าแบบเลขทศนิยม Float 32 Bit จัดเรียงข้อมูลแบบ Big-endian โดยจะแบ่งค่า 32 Bit แยกเป็นแบบ 16 มิติ 2 ชุด โดยส่ง LSB Word ผ่าน Input Register[3] และส่ง MSB Word ผ่าน Input Register[4]
- Holding Register[5:6] ใช้กำหนดค่าชดเชยความผิดพลาดของความชื้น เป็นค่าแบบเลขทศนิยม Float 32 Bit จัดเรียงข้อมูลแบบ Big-endian โดยจะแบ่งค่า 32 Bit แยกเป็นแบบ 16 มิติ 2 ชุด โดยส่ง LSB Word ผ่าน Input Register[5] และส่ง MSB Word ผ่าน Input Register[6]

Input Register (16 Bit Read Only Register)		
Address	Function	Data Type
0000	Firmware Version	Unsigned 16 Bit
0001	Temperature Sensor Value (Signed/10)	Signed 16 Bit
0002	Humidity Sensor Value (Signed/10)	Signed 16 Bit
0003	Temperature Sensor Value (Float)	Float 32Bit (Big-endian)
0004		
0005	Humidity Sensor Value (Float)	Float 32 Bit (Big-endian)
0006		

- Input Register[0] ใช้แสดงค่า Firmware Version ของบอร์ด เป็นค่าแบบ Unsigned 16 Bit เช่น 100 หมายถึง Version 1.00
- Input Register[1] ใช้แสดงค่าอุณหภูมิของเซ็นเซอร์ เป็นค่าแบบ Signed 16 Bit ในหน่วยขององศาเซลเซียส โดยจะเป็นค่าแบบเลขจำนวนเต็ม Signed/10 เมื่อต้องการค่าจริงของเซ็นเซอร์ต้องหารด้วย 10 ก่อน เช่น 25.6 องศาเซลเซียส จะส่งค่าผ่าน Input Register[1] เป็น 256
- Input Register[2] ใช้แสดงค่าความชื้นของเซ็นเซอร์ เป็นค่าแบบ Signed 16 Bit ในหน่วยของ %RH โดยจะเป็นค่าแบบเลขจำนวนเต็ม Signed/10 เมื่อต้องการค่าจริงของเซ็นเซอร์ต้องหารด้วย 10 ก่อน เช่น 56.1%RH จะส่งค่าผ่าน Input Register[2] เป็น 561
- Input Register[3:4] เป็นค่าอุณหภูมิของเซ็นเซอร์ แบบเลขทศนิยม Float 32 Bit จัดเรียงข้อมูลแบบ Big-endian โดยจะแบ่งค่า 32 Bit แยกเป็นแบบ 16 มิติ 2 ชุด โดยส่ง LSB Word ผ่าน Input Register[3] และส่ง MSB Word ผ่าน Input Register[4]
- Input Register[5:6] เป็นค่าความชื้นของเซ็นเซอร์ แบบเลขทศนิยม Float 32 Bit จัดเรียงข้อมูลแบบ Big-endian โดยจะแบ่งค่า 32 Bit แยกเป็นแบบ 16 มิติ 2 ชุด โดยส่ง LSB Word ผ่าน Input Register[5] และส่ง MSB Word ผ่าน Input Register[6]

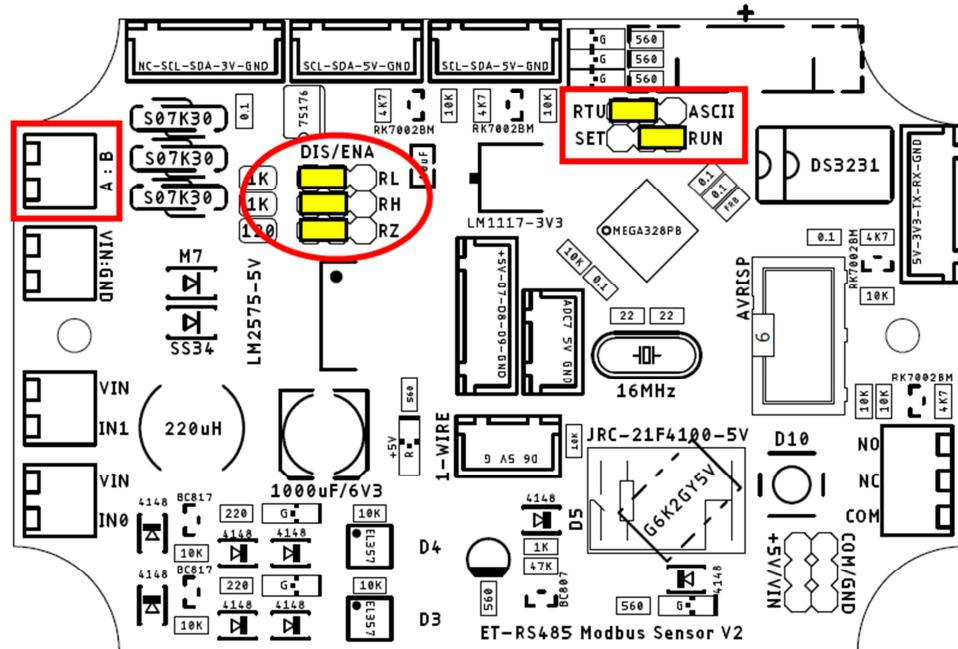
โครงสร้างทางสาร์ดแวร์



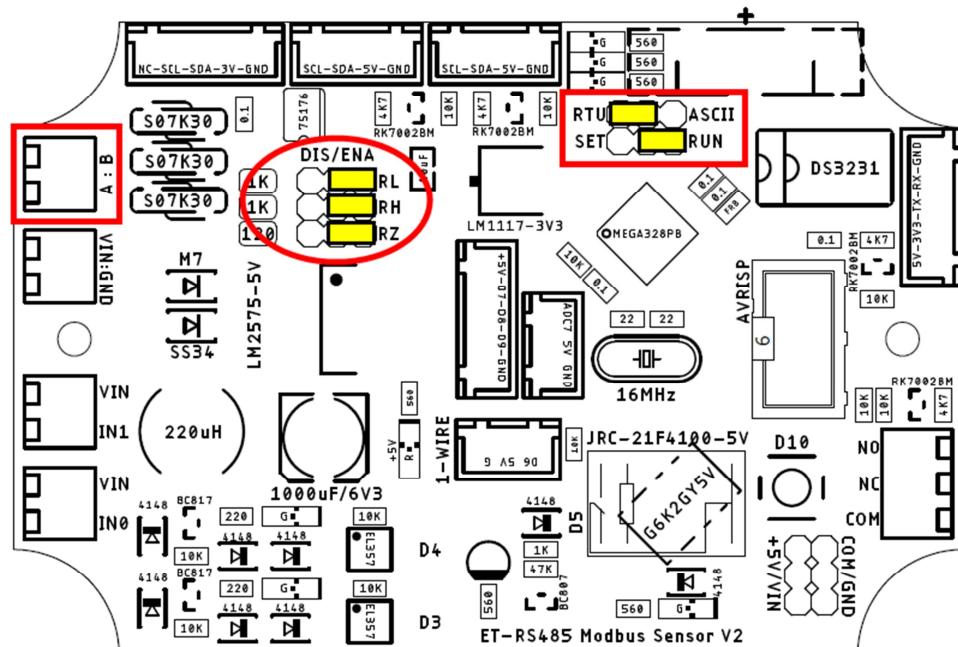
รูปแสดง โครงสร้างของบอร์ด ET-Modbus RTU SHT31 BOX V2

การกำหนด Jumper สำหรับ RS485

ในการกำหนดการเชื่อมต่อ RS485 ของบอร์ดจะสามารถกำหนดได้ 2 แบบ คือ บอร์ดสำหรับติดตั้งใช้งานในกล่องระบบบัส และ บอร์ดสำหรับติดตั้งในตู้แขวนท้ายสุดของระบบบัส โดยกำหนดจาก Jumper สำหรับเลือก Disable Terminate Resistance และ Enable Terminate Resistance



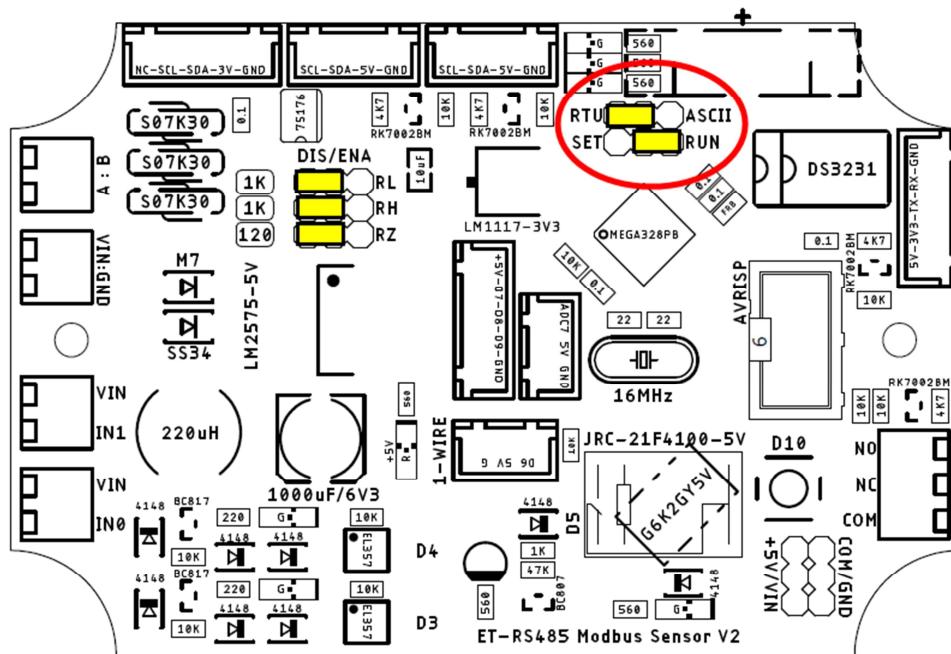
รูปแสดง Jumper RS485 สำหรับบอร์ดที่ไม่ติดตั้งในบัส (Disable Terminate Resistance)



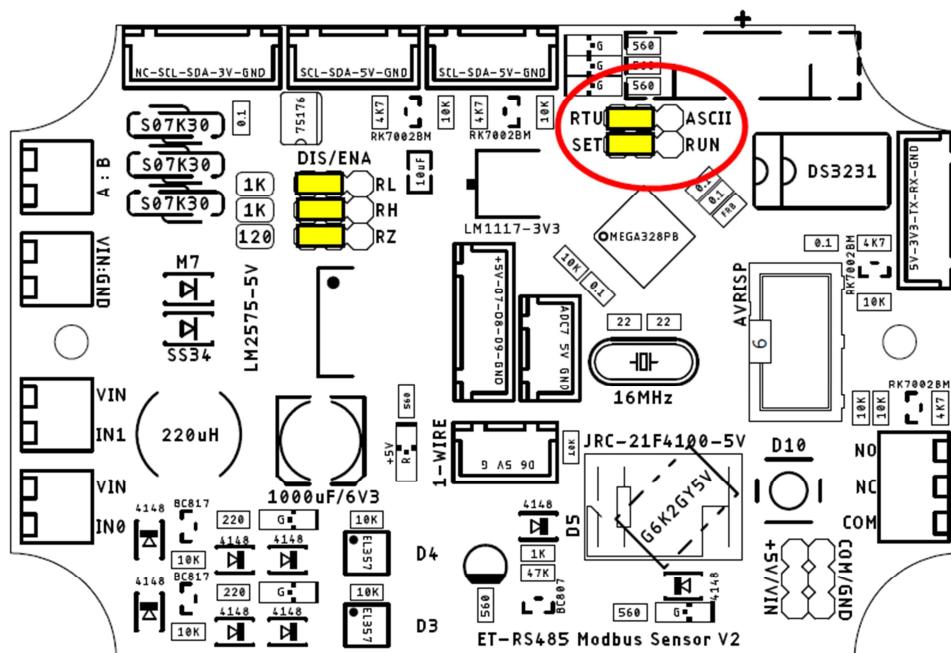
รูปแสดง Jumper RS485 สำหรับบอร์ดสุดท้ายในบัส (Enable Terminate Resistance)

การกำหนด Jumper สำหรับ Setup Slave ID

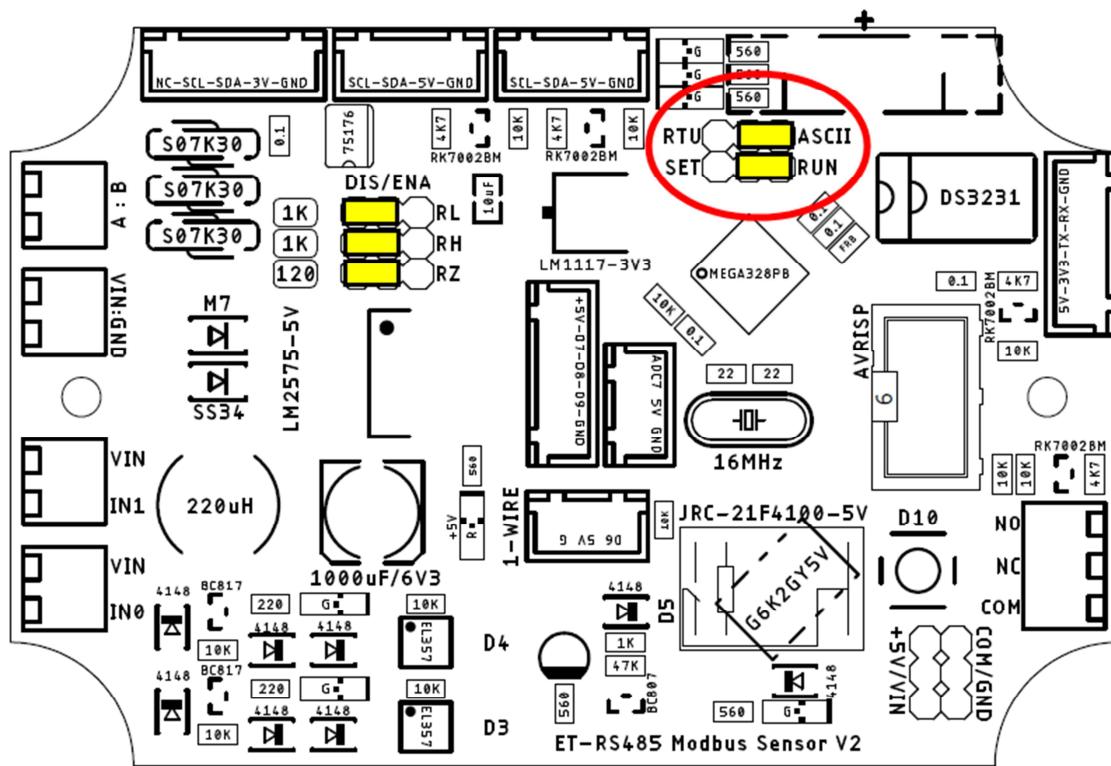
บอร์ด ET-Modbus RTU SHT31 BOX V2 จะใช้การ กำหนดค่า Slave ID ผ่านทางคำสั่งในบัส เพื่อป้องกันความผิดพลาดการสั่งเปลี่ยน Slave ID โดยไม่ต้องใช้ในขณะที่เชื่อมต่อนอร์ดจำนวนมากกว่า 1บอร์ด ร่วมกัน ไว้ในบัส จึงใช้ Jumper SET/RUN สำหรับ Enable และ Disable การสั่งเปลี่ยนค่า Slave ID โดยเมื่อกำหนด Jumper = SET จะเป็นการยินยอมให้สั่งเปลี่ยนค่า Slave ID ได้ และ RUN จะไม่ยินยอมให้สั่งเปลี่ยนค่า Slave ID



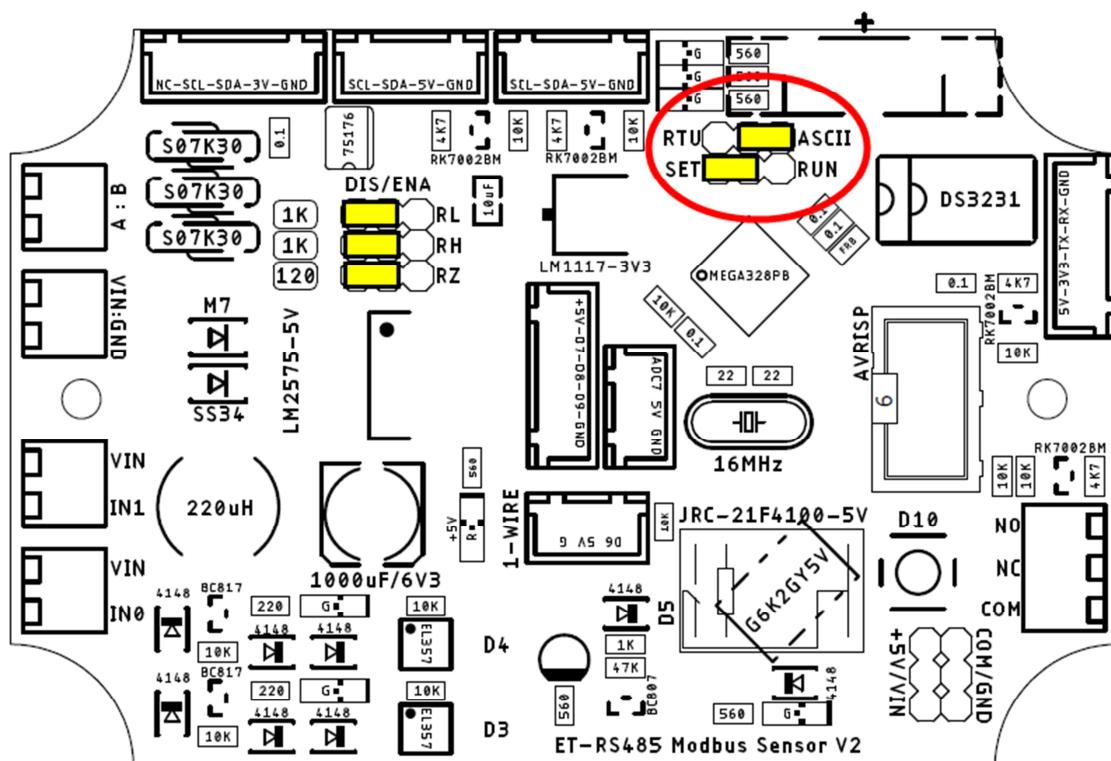
รูปแสดง Jumper สำหรับ Run Modbus Mode เพื่อป้องกันการสั่งเปลี่ยน Slave ID



รูปแสดง Jumper สำหรับ Setup Modbus Mode เพื่อยินยอมให้ Setup Slave ID



รูปแสดง Jumper สำหรับ Run ASCII Mode เพื่อป้องกันการสั่งเปลี่ยน Slave ID

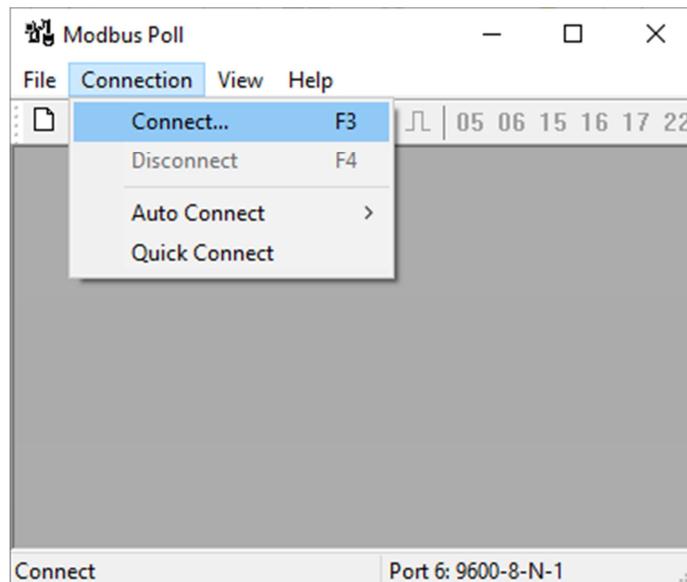


รูปแสดง Jumper สำหรับ Setup ASCII Mode เพื่อ ยินยอมให้ Setup Slave ID

การทดสอบการทำงานของบอร์ดโดยใช้ Modbus Poll

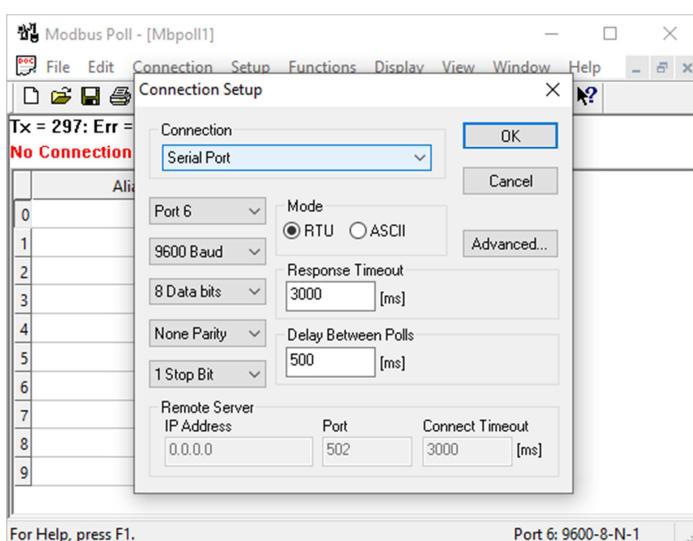
Modbus Poll เป็นโปรแกรมสำเร็จรูปสำหรับใช้ติดต่อสื่อสารระหว่างคอมพิวเตอร์ PC กับอุปกรณ์ที่ใช้การสื่อสารผ่าน Modbus RTU ทางพอร์ตสื่อสารแบบ RS485 สามารถใช้ทดสอบ อ่าน เขียน ข้อมูลบน Protocol โดยไม่ต้องเสียเวลาเขียนโปรแกรมเอง แม้ว่าจะเป็นโปรแกรมมีลิขสิทธิ์แต่ก็สามารถใช้งานแบบ Demo ภายใต้ข้อจำกัดได้เป็นอย่างดี สามารถ Download มาทดลองใช้ได้จาก <https://www.modbustools.com/> ตัวอย่างการกำหนดค่า Modbus Poll สำหรับเชื่อมต่อ สั่งงานบอร์ด ET-Modbus RTU SHT31 BOX V2 ทำได้ดังนี้

1 ไปที่เมนู Connection → Connect...



เลือกกำหนดพารามิเตอร์สำหรับการเชื่อมต่อ RS485 ดังนี้

- Connection เลือกเป็น Serial Port
- Port เลือกกำหนดหมายเลข Com Port ที่ใช้เชื่อมต่อ RS485 กับบอร์ดตามค่าที่ใช้จริง
- Baudrate เลือกเป็น 9600 Baud
- Data เลือกเป็น 8 Bit Data
- Parity เลือกเป็น None Parity
- Stop Bit เลือกเป็น 1 Stop Bit

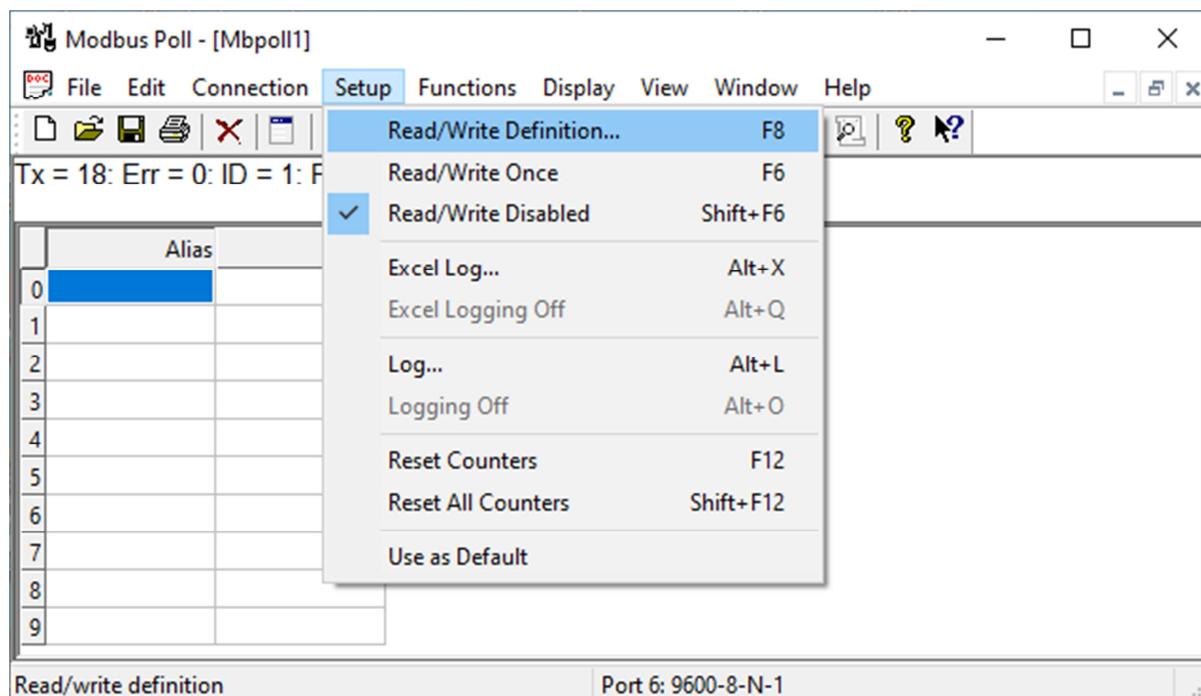


- Mode เลือกเป็น RTU
- Response Timeout กำหนดเป็น 3000mS
- Delay Between Poll กำหนดเป็น 500mS

การสร้างรูปแบบการอ่านเขียนข้อมูลกับบอร์ด

ในการกำหนดให้โปรแกรม Modbus Poll ทำการสื่อสารกับบอร์ด ET-Modbus RTU SHT31 BOX V2 สามารถทำได้โดย เลือกเมนู Setup → Read/Write Definition... จากนั้นเลือกกำหนดรูปแบบการอ่านเขียนข้อมูลกับ Modbus Register ตามที่ต้องการ โดยในกรณีของบอร์ด ET-Modbus RTU SHT31 BOX V2 จะรองรับรีจิสเตอร์ 2 ชนิดคือ

- Holding Register ใช้สำหรับ อ่าน และ เขียน เพื่อกำหนดค่า Slave ID และ ค่าชดเชยค่าความคลาดเคลื่อนของค่าอุณหภูมิและความชื้นที่อ่านได้จากเซ็นเซอร์
 - Slave ID
 - Temperature Adjust แบบเลขจำนวนเต็ม(Signed/10)
 - Humidity Adjust แบบเลขจำนวนเต็ม(Signed/10)
 - Temperature Adjust แบบเลขทศนิยม(Float 32Bit)
 - Humidity Adjust แบบเลขทศนิยม(Float 32Bit)
- Input Register ใช้สำหรับ อ่าน ค่าและแสดงผลค่าพารามิเตอร์ต่างๆของบอร์ด ซึ่งสามารถอ่านได้อย่างเดียวไม่สามารถเปลี่ยนแปลงแก้ไขได้ เช่น
 - Firmware Version ของบอร์ด
 - Temperature Sensor Value แบบเลขจำนวนเต็ม (Signed/10)
 - Humidity Sensor Value แบบเลขจำนวนเต็ม (Signed/10)
 - Temperature Sensor Value แบบเลขทศนิยม (Float 32Bit)
 - Humidity Sensor Value แบบเลขทศนิยม (Float 32Bit)

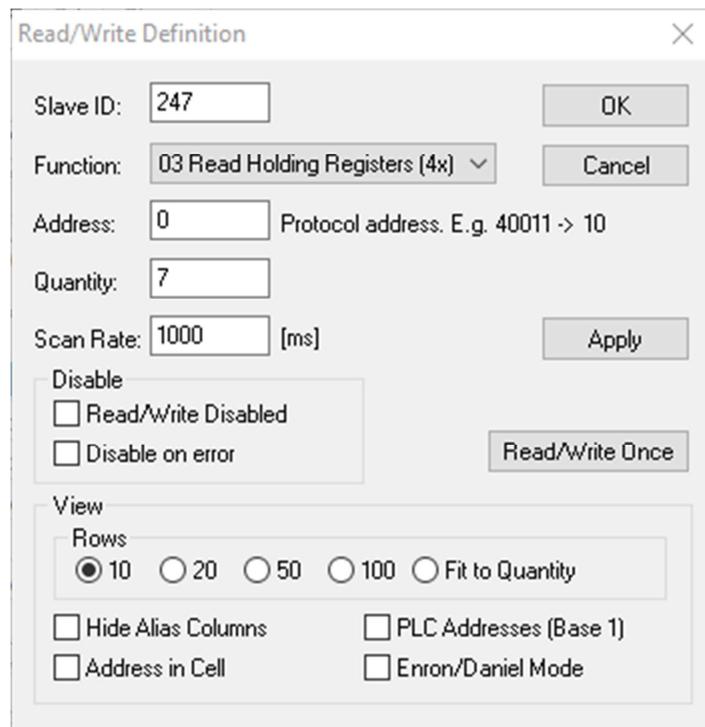


รูปแสดง การกำหนดรูปแบบการอ่านเขียนข้อมูลของโปรแกรม Modbus Poll กับบอร์ด

การกำหนดการ อ่าน/เขียน Holding Register

การกำหนดให้โปรแกรม Modbus Poll อ่าน/เขียน Holding Register ของบอร์ด ET-Modbus RTU SHT31 BOX V2 ทำได้โดยเลือกที่เมนู Setup → Read/Write Definition... และเลือกกำหนดค่าให้โปรแกรมดังนี้

- กำหนด Slave ID ให้ตรงกับหมายเลข Slave ID ของบอร์ด
- กำหนด Function = 03 Read Holding Register(4x)
- กำหนด Address = 0 (เริ่มต้นอ่าน Holding Register ตำแหน่ง 0x0000)
- กำหนด Quantity = 7 (กำหนดจำนวน Modbus Register ในการอ่าน 4ตำแหน่ง)
- กำหนด Scan Rate = 100mS



The table displays the following data:

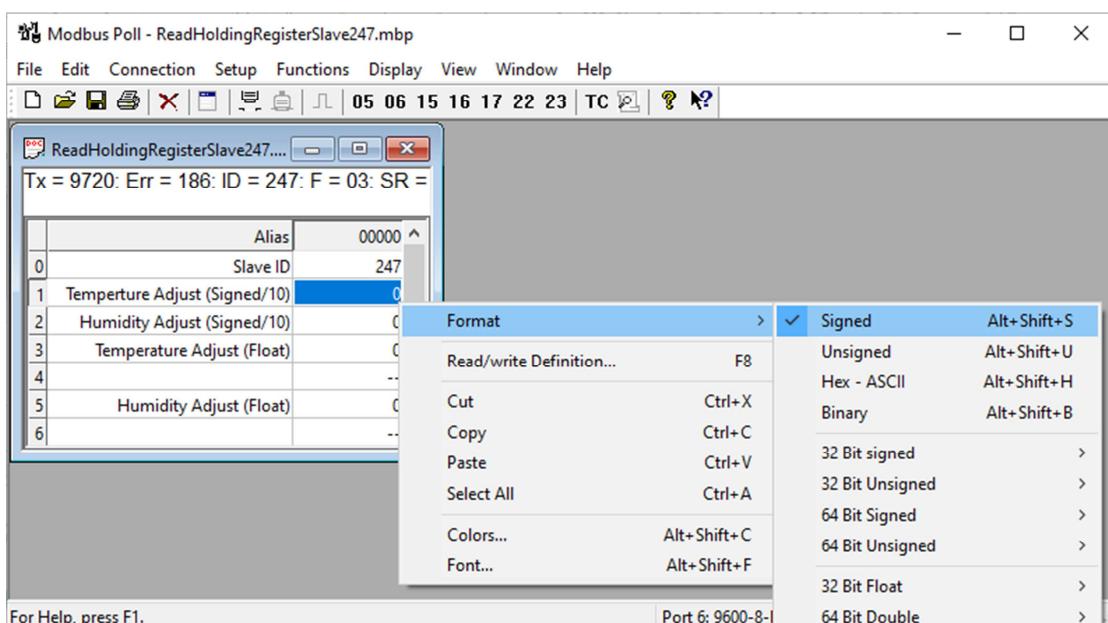
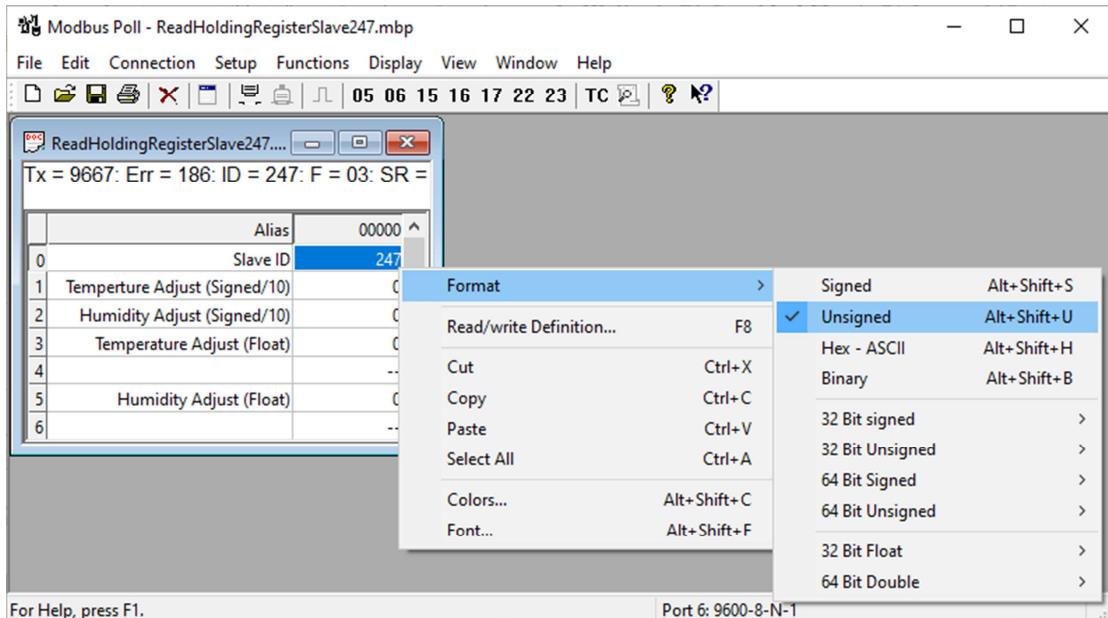
	Alias	00000
0	Slave ID	247
1	Temperture Adjust (Signed/10)	0
2	Humidity Adjust (Signed/10)	0
3	Temperature Adjust (Float)	0
4		--
5	Humidity Adjust (Float)	0
6		--

รูปแสดงการใช้ Modbus Poll สำหรับ อ่าน/เขียน Holding Register

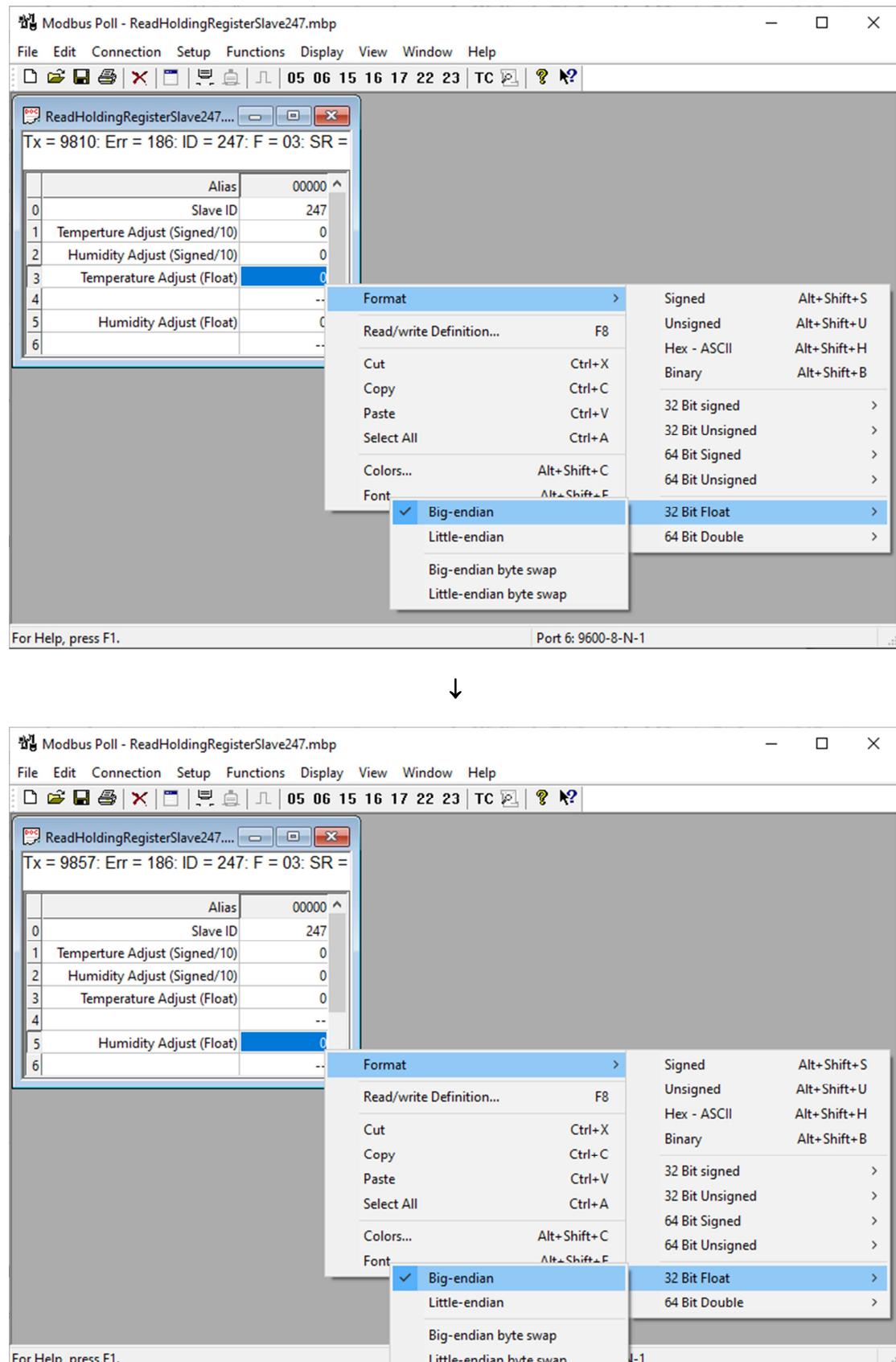
การกำหนดค่าการแสดงผลของ Holding Register

กำหนดรูปแบบการแสดงผลข้อมูลของโปรแกรม Modbus Poll ทำได้โดยการคลิกเมาส์ที่ตำแหน่งของช่องแสดงค่าของ Holding Register ตำแหน่งที่ต้องการแล้วคลิกขวาที่เม้าส์ เลือก Format = รูปแบบการแสดงค่าตามรูปแบบชนิดของตัวเลขที่ต้องการ โดยในกรณีของบอร์ด เลือกกำหนดดังนี้

- ที่ช่องแสดงค่า Holding ตำแหน่ง 0000 เลือกการแสดงผลเป็น Unsigned หรือ เลขจำนวนเต็ม
- ที่ช่องแสดงค่า Holding ตำแหน่ง 0001,0002 เลือกการแสดงผลเป็น Signed หรือ เลขจำนวนเต็ม
- ที่ช่องแสดงค่า Holding ตำแหน่ง 0003,0005 เลือกการแสดงผลเป็น 32 Bit Float Big-endian



คู่มือการใช้งาน ET-Modbus RTU SHT31 BOX V2

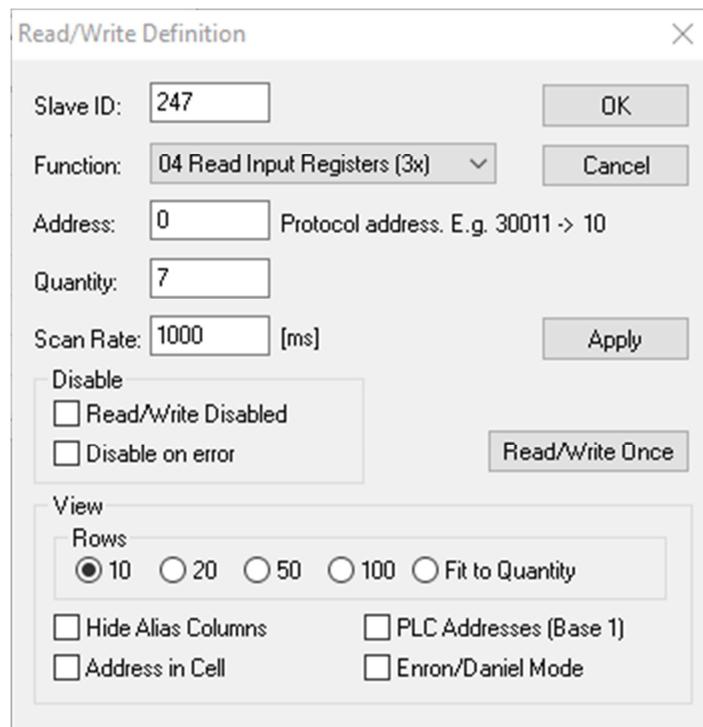


รูปแสดงการกำหนดให้โปรแกรม Modbus Poll แสดงค่าของ Holding Register ของบอร์ด

การกำหนดการ อ่าน/เขียน Input Register

การกำหนดให้โปรแกรม Modbus Poll อ่าน Input Register ของบอร์ด ET-Modbus RTU SHT31 BOX V2 ทำได้โดยเลือกที่เมนู Setup → Read/Write Definition... และเลือกกำหนดค่าให้โปรแกรมดังนี้

- กำหนด Slave ID ให้ตรงกับหมายเลข Slave ID ของบอร์ด
- กำหนด Function = 04 Read Input Register(3x)
- กำหนด Address = 0 (เริ่มต้นอ่าน Input Register ตำแหน่ง 0x0000)
- กำหนด Quantity = 7 (กำหนดจำนวน Modbus Register ในการอ่าน 7ตำแหน่ง)
- กำหนด Scan Rate = 100mS



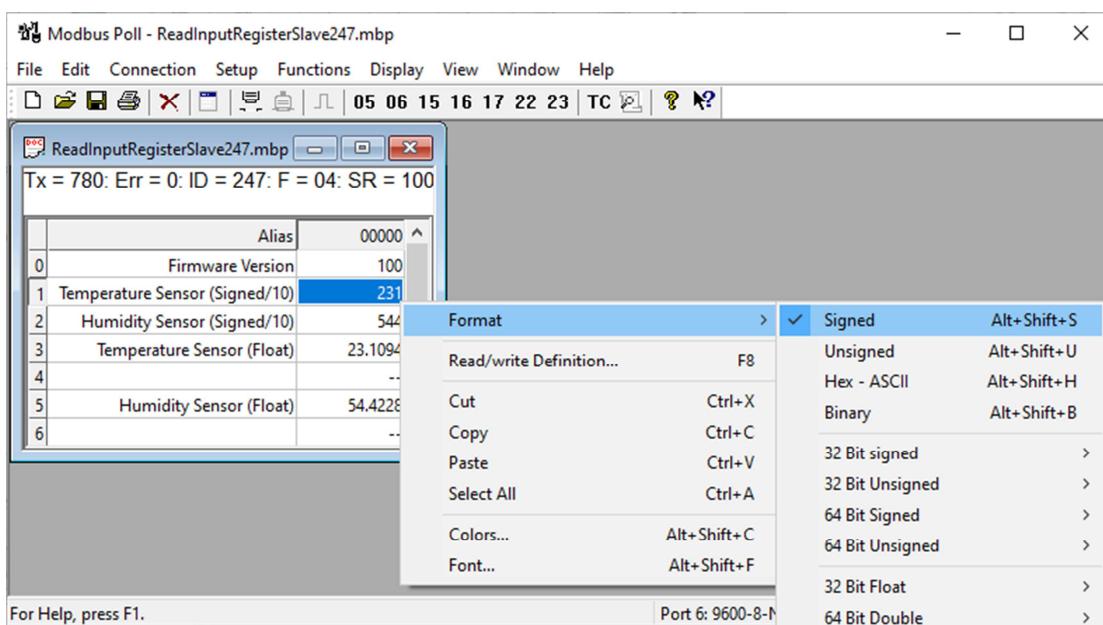
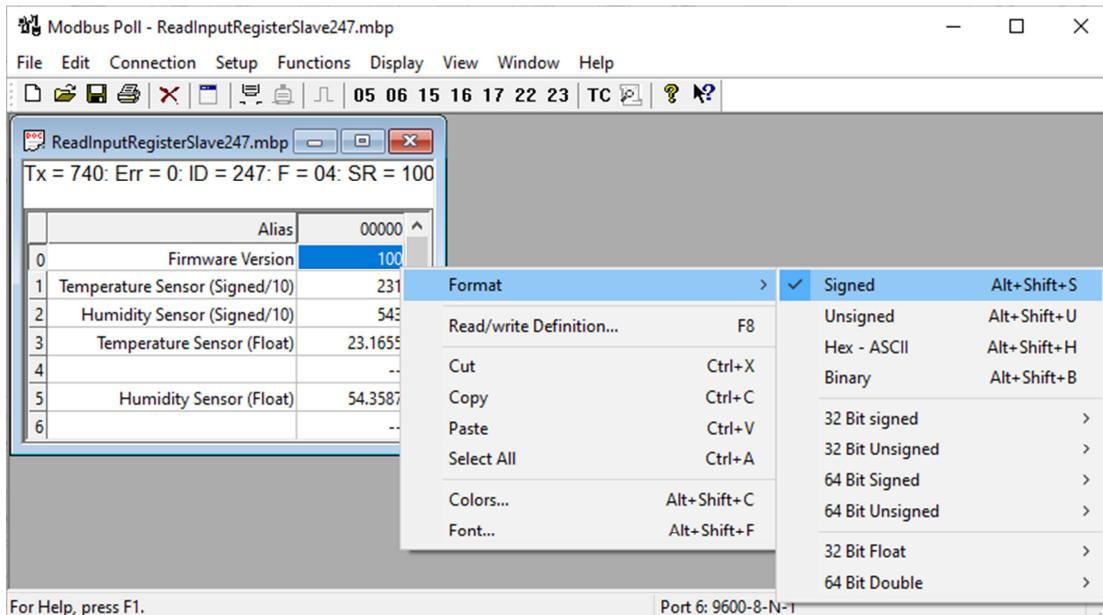
	Alias	
0	Firmware Version	100
1	Temperature Sensor (Signed/10)	231
2	Humidity Sensor (Signed/10)	546
3	Temperature Sensor (Float)	23.1228
4		--
5	Humidity Sensor (Float)	54.6288
6		--

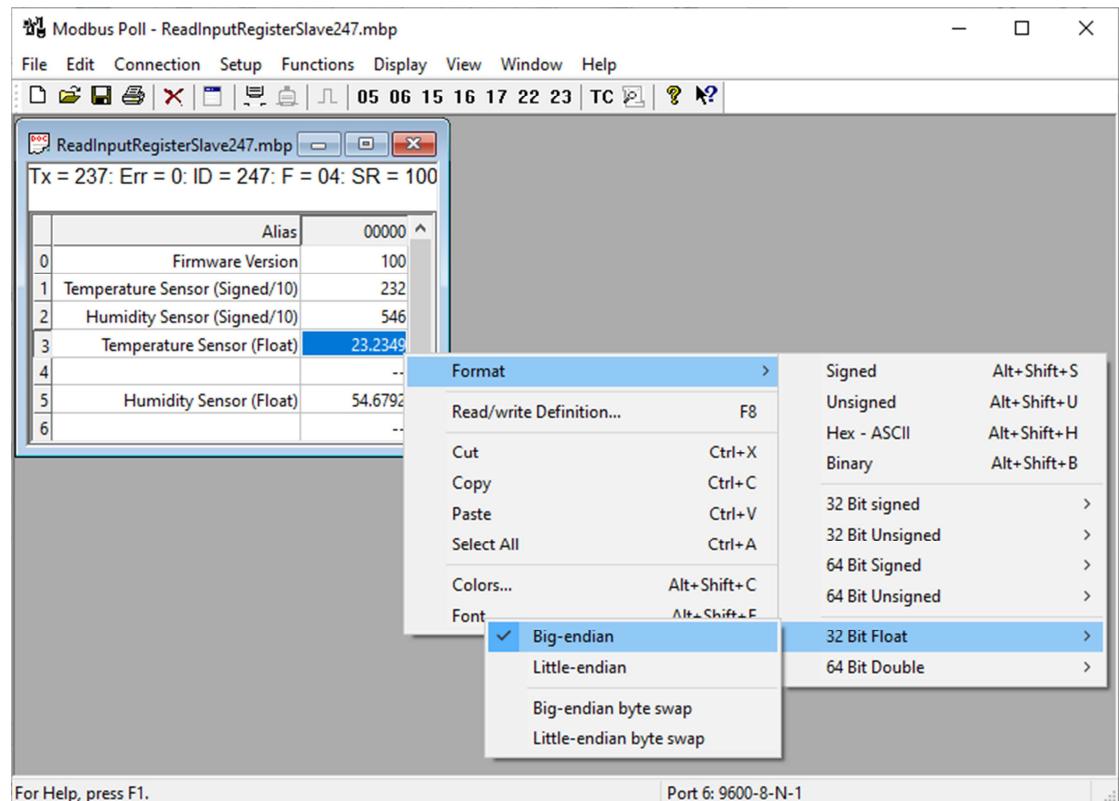
รูปแสดงการใช้ Modbus Poll สำหรับ อ่าน Input Register

การกำหนดค่าการแสดงผลของ Input Register

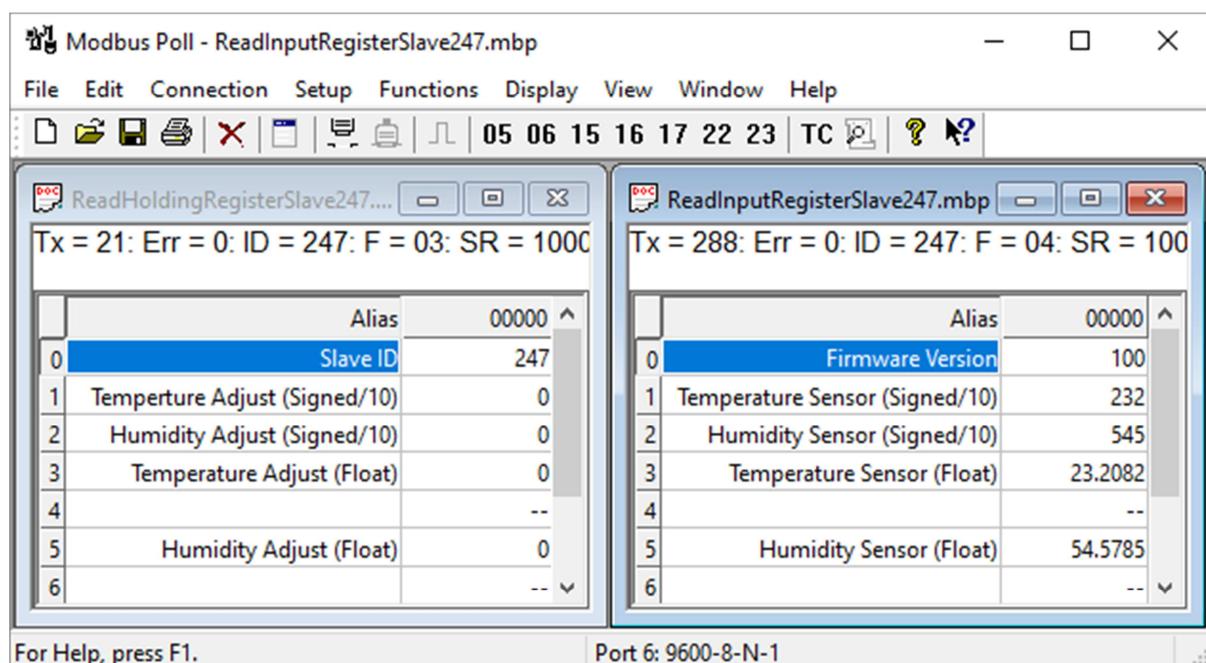
กำหนดรูปแบบการแสดงผลข้อมูลของโปรแกรม Modbus Poll ทำได้โดยการคลิกมาส์ที่ตำแหน่งของช่องแสดงค่าของ Input Register ตัวแทนที่ต้องการแล้วคลิกขวาที่มาส์ เลือก Format = รูปแบบการแสดงค่าตามรูปแบบชนิดของตัวเลขที่ต้องการ โดยในกรณีของบอร์ด ET-Modbus RTU SHT31 BOX V2 เลือกกำหนดดังนี้

- ที่ช่องแสดงค่า Input ตำแหน่ง 0000 เลือกการแสดงผลเป็น Unsigned
- ที่ช่องแสดงค่า Input ตำแหน่ง 0001 และ 0002 เลือกการแสดงผลเป็น Signed
- ที่ช่องแสดงค่า Input ตำแหน่ง 0003 และ 0005 เลือกการแสดงผลเป็น 32 Bit Float Big-endian





รูปแสดง การกำหนดค่าแสดงผล Input Register ของ Modbus Poll

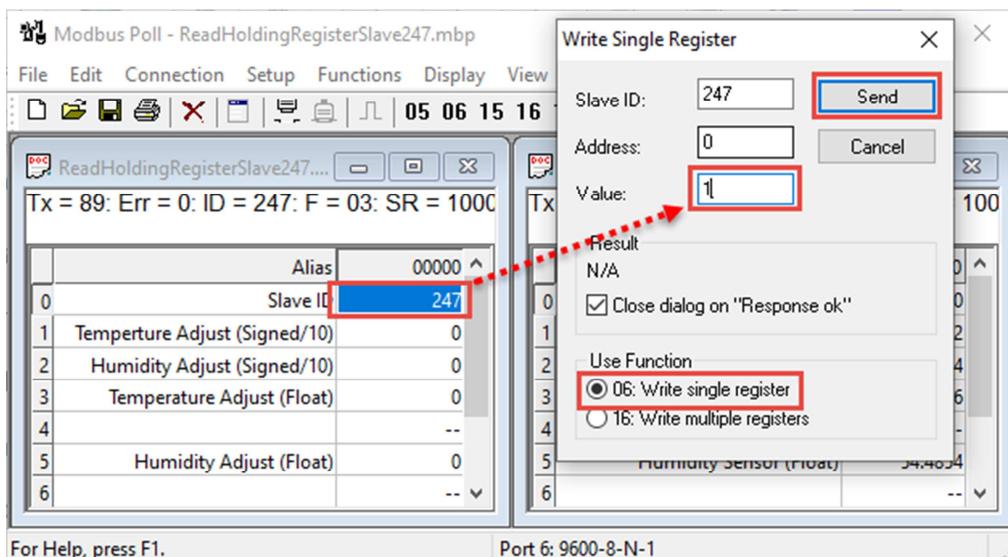


รูปแสดง ตัวอย่าง การแสดงค่า Holding และ Input ด้วย Modbus Poll

ตัวอย่าง การกำหนดค่า Slave ID ให้กับบอร์ด

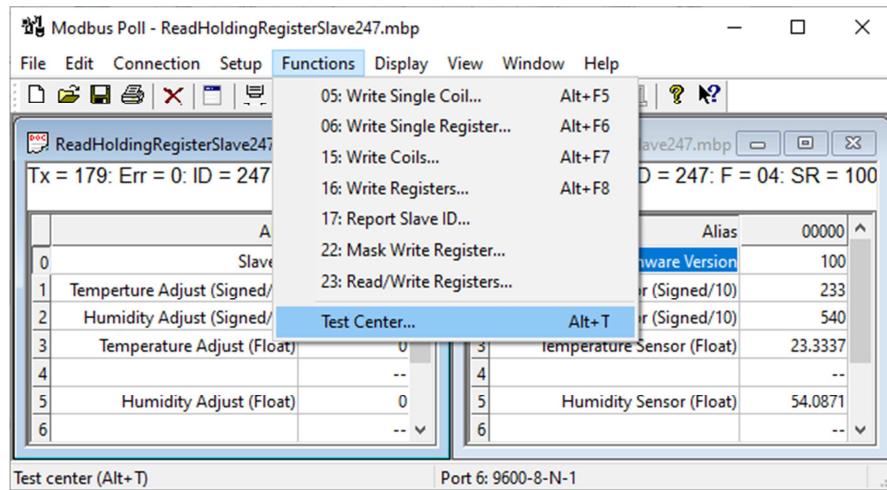
ตามปกติแล้วบอร์ดจะถูกกำหนดค่า Slave ID ให้มีค่าเป็น 247 ไว้ แต่เมื่อต้องการต่อบอร์ดร่วมกันในบัส RS485 มากกว่า 1 บอร์ด จะเป็นต้องกำหนดหมายเลข Slave ID ให้บอร์ดมีหมายเลขที่ไม่ซ้ำกัน แต่ในกรณีนี้ต้องทำการกำหนด Jumper(SET/RUN) ของบอร์ดไว้ที่ตำแหน่ง SET และควรต่อบอร์ดไว้ในบัส RS485 เพียงบอร์ดเดียวเท่านั้น ไม่เช่นนั้นแล้วบอร์ดทุกบอร์ดที่เลือก Jumper(SET/RUN) ไว้ที่ SET จะถูกกำหนดให้มีค่า Slave ID เหมือนกันหมดทุกบอร์ดและเมื่อทำการกำหนดค่า Slave ID เรียบร้อยแล้วควรเลือก Jumper กลับมาไว้ในตำแหน่ง RUN ด้วยเพื่อป้องกันความผิดพลาดจากการสั่งเขียนค่า Holding Register ตำแหน่ง 0x0000 โดยไม่ตั้งใจได้ ในกรณีที่ทราบหมายเลข Slave ID ของบอร์ด สามารถทำการกำหนดค่าหมายเลข Slave ID ใหม่ ให้กับบอร์ด โดยการ สั่งเขียนค่า หมายเลข Slave ID ใหม่ที่ต้องการ(1..247) ไปยัง Holding Register ตำแหน่ง Address 0(0x0000) ของบอร์ดที่ต้องการ และในกรณีที่ไม่ทราบหมายเลข Slave ID เดิมของบอร์ด สามารถกำหนดหมายเลข Slave ID ของบอร์ดได้โดยใช้การอ้างหมายเลข Slave ID Broadcast (0) และสั่งเขียนค่า หมายเลข Slave ID ใหม่ที่ต้องการ(1..247) ไปยัง Holding Register ตำแหน่ง Address 0(0x0000) ของบอร์ด ในกรณีที่ทราบหมายเลข Slave ID ปัจจุบันของบอร์ด สามารถใช้ Function 0x06(Write Holding Register) เพื่อกำหนดค่าให้กับ Holding Register ตำแหน่ง Address 0x0000 ด้วยค่า 1-247 ได้ทันทีตามต้องการ หรือ Double Click เม้าส์ที่ช่องแสดงค่า Slave ID ของ Holding Register ตำแหน่ง 0x0000 ซึ่งจะปรากฏช่องหน้าต่างสำหรับกำหนดค่า Slave ID ให้แก่ไขค่าได้ทันที ในขั้นตอนนี้สามารถแก้ไขค่า Slave ID ใหม่ให้กับบอร์ดได้โดยเข้าไปแก้ไขค่าตัวเลขที่ช่อง Value ให้มีค่าระหว่าง 1-247 ได้ตามต้องการแล้วเลือก Send ดังตัวอย่าง

- กำหนดหมายเลข Slave ID = 247 (หรืออื่นๆตามค่า Slave ID ของบอร์ด)
- กำหนด Address ของรีจิสเตอร์ที่จะเขียนเป็น = 0 (Holding[0] = Slave ID)
- กำหนด Value = 1(กำหนด Slave ID ใหม่ = 1)
- เลือก Send

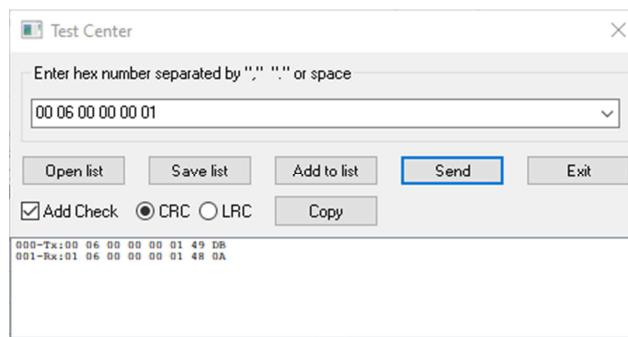


ขึ้นหลังจากสั่งค่าสั่ง Send และ ถ้าทุกอย่างถูกต้องหมายเลข Slave ID ของบอร์ดจะถูกเปลี่ยนใหม่ทันที ต้องทำการกำหนดค่า Slave ID ในโปรแกรม Modbus Poll เป็นค่าใหม่เพื่อติดต่อกับบอร์ดด้วย

ในกรณีที่ไม่ทราบหมายเลข Slave ID ของบอร์ดจะไม่สามารถใช้ Function 0x06 (Write Holding Register) ในการกำหนดค่า Slave ID ได้โดยตรง เนื่องจากโปรแกรม Modbus Poll ไม่รองรับการส่งข้อมูลแบบ Broadcast แต่เราสามารถใช้การสั่งเฟรมคำสั่งเพื่อส่งค่าไปยังบอร์ดได้เอง โดยไปที่เมนู Functions → Test Center...



ให้ทำการป้อนค่าตัวเลขเป็น เลขฐาน 16 จำนวน 6 ใบที่ แต่ละใบตัวบ่งແยກตัวบylevel ของหมายเลขเว้นวรรค โดย ใบที่แรก กำหนดเป็นค่า Slave ID Broadcast = 0x00 แต่พิมพ์เพียง 00 และใบที่ 2 กำหนดเป็นค่าฟังก์ชัน (Write Single Holding Register) = 06 ใบที่ 3 และ 4 กำหนดเป็นค่า Address ของ Holding Register ตำแหน่งที่ 00 = 00 00 ส่วนใบที่ 5 และ 6 กำหนดเป็นค่าหมายเลข Slave ID ใหม่ที่ต้องการ ซึ่งต้องมีค่าระหว่าง 1 ถึง 247 (00 01 ถึง 00 F7) ส่วนค่า CRC ให้โปรแกรมคำนวณและเพิ่มให้อัตโนมัติโดยให้ทำการ Enable Check Box ที่ Add Check และ CRC ดังตัวอย่างข้างกำหนดให้หมายเลข Slave ID = **01** (00 06 00 00 00 01) ซึ่งหลังจากส่งคำสั่ง Send และ ถ้าทุกอย่างถูกต้องหมายเลข Slave ID ของบอร์ดจะถูกเปลี่ยนใหม่ทันที ซึ่งเราต้องทำการกำหนดค่า Slave ID ในโปรแกรม Modbus Poll เป็นค่าใหม่เพื่อติดต่อกับบอร์ดด้วย



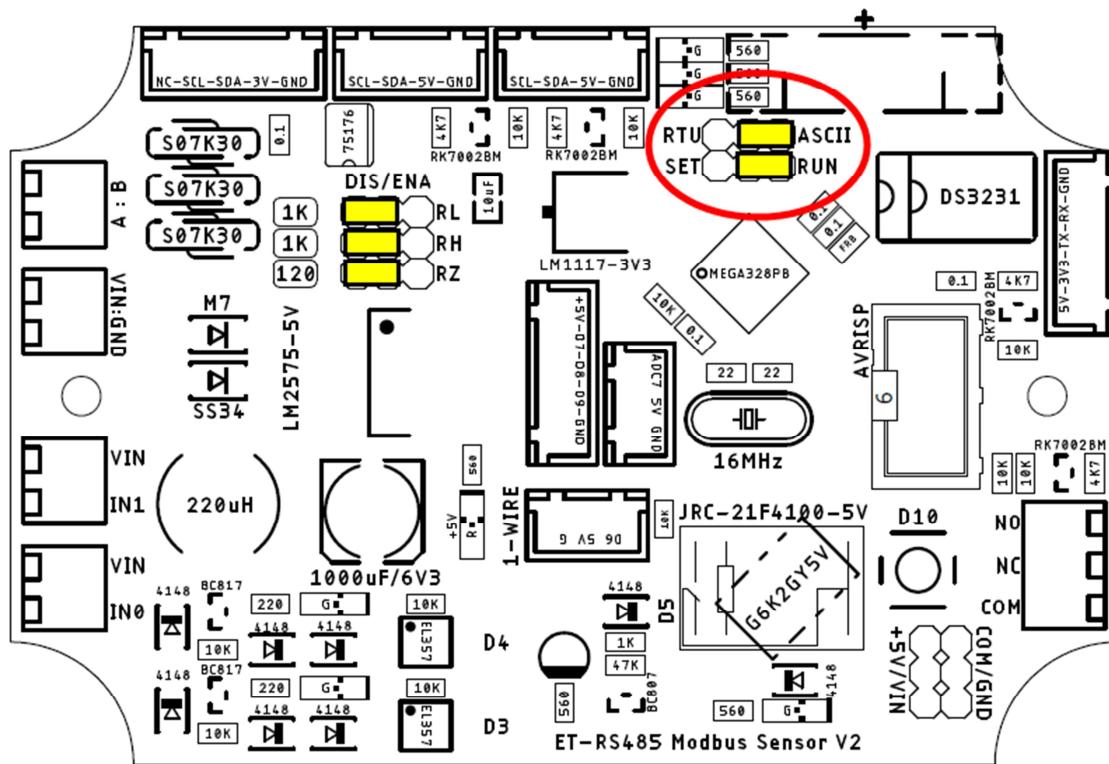
รูปแสดงตัวอย่าง การป้อนเฟรมคำสั่ง สำหรับกำหนดค่า Slave ID = **01** (00 06 00 00 00 01)

ส่งคำสั่ง : **00 06 00 00 00 01 49 DB**

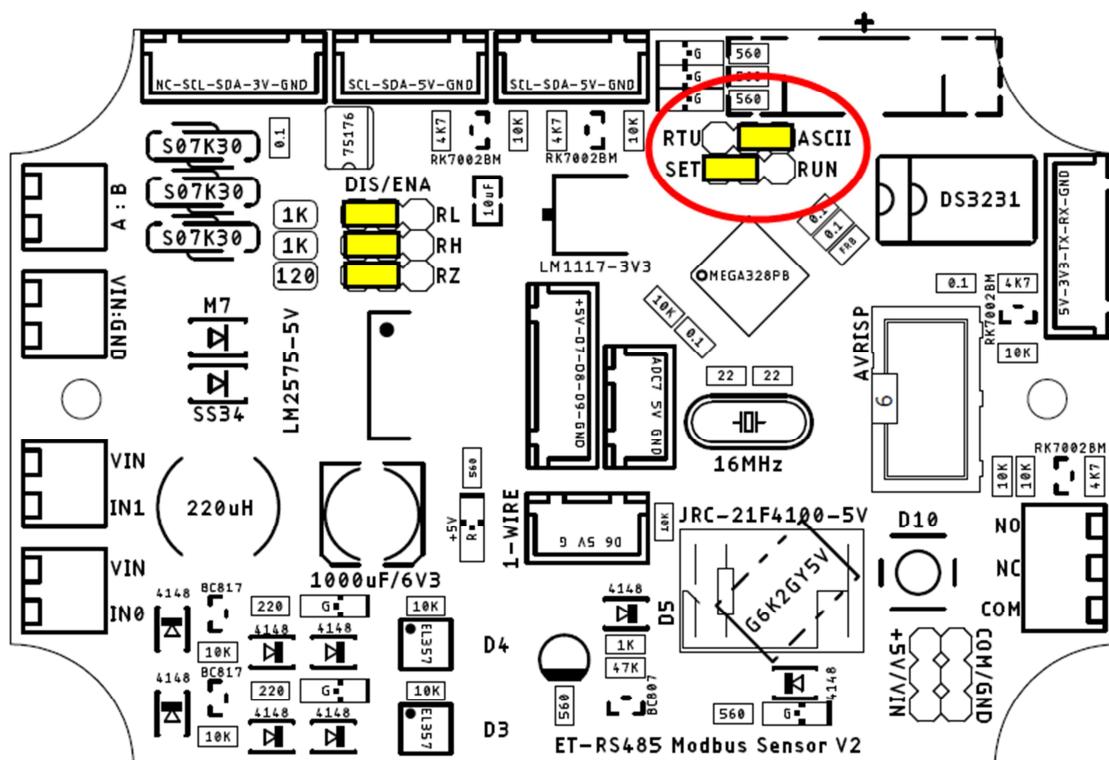
ตอบรับ : **01 06 00 00 00 01 48 0A**

ซึ่งหลังจากส่งคำสั่ง Send และ ถ้าทุกอย่างถูกต้องหมายเลข Slave ID ของบอร์ดจะถูกเปลี่ยนใหม่ทันที ต้องทำการกำหนดค่า Slave ID ในโปรแกรม Modbus Poll เป็นค่าใหม่เพื่อติดต่อกับบอร์ดด้วย

การใช้งานในโหมด ASCII Command



รูปแสดง Jumper ของ ASCII Command แบบ Run Mode เพื่อป้องกันการลั่งเปลี่ยน Slave ID



รูปแสดง Jumper ของ ASCII Command แบบ Setup Mode สำหรับ Setup Slave ID

การสื่อสารใน ASCII Command Mode

ในโหมดการสื่อสารแบบ ASCII Command Mode จะใช้อักขระของตัวเลข 0-9 และ อักขระของเครื่องหมาย [*][:][,] [=][+][-][.][/][?][<CR>] ประกอบกันเป็นชุดคำสั่ง โดยใช้อักขระของเครื่องหมาย [*] เป็นรหัสเริ่มต้นของเฟรมคำสั่ง และสิ้นสุดเฟรมคำสั่งด้วย [<CR>] หรือ Enter หรือ 0x0D หรือ 13(ฐาน10) โดยมีเครื่องหมาย [:][,][=] เป็นตัวแบ่งแยกฟิลด์ ต่างๆ โดยมีรูปแบบคำสั่ง 2รูปแบบ ดังนี้

- RS485 Half Duplex
- Baudrate 9600bps
- Data 8Bit
- None Parity
- 1 Stop Bit

กลุ่มคำสั่งอ่านค่าจากรีจิสเตอร์

[*][ID][:][FUNC][,][ADDR0][<CR>]

- ID เป็นค่าหมายเลข Slave ID เป็นเลขจำนวนเต็มมีค่าระหว่าง 1 ถึง 247
- FUNC เป็นค่า Function ของคำสั่งมีค่า
 - 3(Read Holding Register)
 - 4(Read Input Register)
- ADDR0 ใช้กำหนดตำแหน่งรีจิสเตอร์ที่ต้องการอ่าน

กลุ่มคำสั่งเขียนค่าให้รีจิสเตอร์

[*][ID][:][FUNC][,][ADDR0][=][VALUE0][<CR>]

- ID เป็นค่าหมายเลข Slave ID เป็นเลขจำนวนเต็มมีค่าระหว่าง 1 ถึง 247
- FUNC เป็นค่า Function ของคำสั่งมีค่า
 - 6(Write Holding Register)
- ADDR0 ใช้กำหนดตำแหน่งรีจิสเตอร์ที่ต้องการเขียนค่า
- VALUE0 ใช้กำหนดค่าสำหรับเขียนให้รีจิสเตอร์

รูปแบบคำสั่งการตอบรับจาก Slave

[*][ID][:]FUNC[,][ADDR0][=][VALUE0][<CR>]

- ID เป็นค่าหมายเลข Slave ID เป็นเลขจำนวนเต็มมีค่าระหว่าง 1 ถึง 247
- FUNC เป็นค่า Function ของคำสั่งที่ตอบรับการทำงาน(1..6)
 - 3 Read Holding Register
 - ADDR0
 - 0 = Slave ID
 - VALUE0
 - 1..247 = Slave ID
 - 1 = Temperature Adjust
 - VALUE0
 - 0.0 = No Adjust
 - -40.0 ถึง +125.0 = Adjust Value
 - 2 = Humidity Adjust
 - VALUE0
 - 0.0 = No Adjust
 - -100.0 ถึง +100.0 = Adjust Value
 - 4 Read Input Register
 - ADDR0
 - 0 = Read Firmware Version
 - V1.00
 - 1 = Read Temperature Sensor
 - VALUE0(-40.0 ถึง +125.0 °C)
 - 2 = Read Humidity Sensor
 - VALUE0(0.0 ถึง +100.0 %RH)
 - 6 Write Holding Register
 - ADDR0
 - 0 = Slave ID
 - VALUE0
 - 1..247 = Slave ID
 - 1 = Write Temperature Adjust
 - VALUE0
 - 0.0 = No Alarm
 - -40.0 ถึง +125.0 = Temperature Adjust Value
 - 2 = Write Alarm PM10
 - VALUE0
 - 0.0 = No Adjust
 - -100.0 ถึง +100.0 = Humidity Adjust Value

Firmware ของบอร์ด ET-Modbus RTU SHT31 BOX V2 ใน ASCII Command Mode ได้ออกแบบและกำหนดรีจิสเตอร์ สำหรับใช้เป็นตัวกลางในการติดต่อสั่งงาน ระหว่างผู้ใช้กับอุปกรณ์ต่างๆในบอร์ด จำนวน 2 กลุ่ม การสั่งงาน และการกำหนดเงื่อนไขการทำงาน และการอ่านค่าต่างๆจะกระทำผ่านรีจิสเตอร์ ซึ่งแสดงให้เห็นดังตารางต่อไปนี้

Holding Register(16 Bit Read/Write Register)		
Address	Function	Data Type
0000	Read/Write: Slave ID Code (1...247)	ASCII unsigned(1..247)
0001	Read/Write: Temperature Adjust (0.0,+/-125.0)	ASCII Float
0002	Read/Write: Humidity Adjust (0.0,+/-100.0)	ASCII Float

Input Register(16 Bit Read Only Register)		
Address	Function	Data Type
0000	Read: Firmware Version (100 : V1.00)	ASCII unsigned (100 : 1.00)
0001	Read: Temperature Sensor (-40.0 ถึง +125.0)	ASCII Float (-40 ถึง 125.0 °C)
0002	Read: Humidity Sensor (0.0 ถึง 100.0)	ASCII Float (0.0 ถึง 100.0 %RH)

ตาราง 1-4 แสดง ตำแหน่ง RS485 Register สำหรับใช้ติดต่อสื่อสารกับอุปกรณ์ต่างๆในบอร์ด

1 การอ่านค่า Holding Register

Holding Register(16 Bit Read/Write Register)		
Address	Function	Data Type
0000	Read/Write: Slave ID Code (1...247)	ASCII unsigned(1..247)
0001	Read/Write: Temperature Adjust (0.0,+/-125.0)	ASCII Float
0002	Read/Write: Humidity Adjust (0.0,+/-100.0)	ASCII Float

การสั่งอ่านค่า Holding Register เป็นการสั่งค่าสถานะของ Configuration ต่างๆที่กำหนดไว้ ว่ามีค่าเป็นเท่าไร โดยจะใช้ Function 3 ในการอ่าน มีรูปแบบคำสั่งการใช้งานดังนี้

รูปแบบคำสั่งการสั่งงานจาก Master

[*][ID][:][FUNC][,][ADDR0][<CR>]

- ID เป็นค่าหมายเลข Slave ID เป็นเลขจำนวนเต็มมีค่าระหว่าง 1 ถึง 247
- FUNC เป็นค่า Function ของคำสั่งมีค่า = 3(Read Holding Register)
- ADDR0 ใช้กำหนดตำแหน่งรีลิสเตอร์สำหรับควบคุมการทำงานของ Relay
 - 0 หมายถึง Slave ID
 - 1 หมายถึง Temperature Adjust
 - 2 หมายถึง Humidity Adjust

รูปแบบคำสั่งการตอบรับจาก Slave

[*][ID][:][FUNC][,][ADDR0][=][VALUE0][<CR>]

- ID เป็นค่าหมายเลข Slave ID เป็นเลขจำนวนเต็มมีค่าระหว่าง 1 ถึง 247
- FUNC เป็นค่า Function ของคำสั่งมีค่า = 3(Read Holding Register)
- ADDR0 ใช้บอกตำแหน่งรีลิสเตอร์ที่สั่งอ่านค่าจาก Master(0..2)
- VALUE0 ใช้แสดงค่าปัจจุบันของ Holding Register ที่สั่งอ่าน

ตัวอย่างคำสั่งอ่านค่า Holding Register

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:3,0<CR>	Master สั่งอ่าน Holding Register[0] จาก Slave1
Slave ตอบรับ	*1:3,0=1<CR>	Holding Register[0] หรือ Slave ID = 1

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:3,1<CR>	Master สั่งอ่าน Holding Register[1] จาก Slave1
Slave ตอบรับ	*1:3,1=0.0<CR>	Holding Register[1] หรือ Temperature Adjust = 0.0

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:3,2<CR>	Master สั่งอ่าน Holding Register[2] จาก Slave1
Slave ตอบรับ	*1:3,2=0.0<CR>	Holding Register[2] หรือ Humidity Adjust = 0.0

ตัวอย่างคำสั่งอ่านค่า Slave ID ของบอร์ด

ค่าหมายเลข Slave ID ของบอร์ด เป็นรหัสประจำตัวของบอร์ด เพื่ออ้างถึงในการสื่อสารกับบอร์ด ตามปกติแล้วจะถูกบรรจุไว้ใน Holding[0] ซึ่งเป็น Holding Register ตำแหน่งที่ 0 ซึ่งเราสามารถใช้รูปแบบคำสั่งสำหรับสั่งอ่าน Holding Register ตามปกติได้ ซึ่งก็ไม่มีความจำเป็น เพราะในรูปแบบของคำสั่ง ต้องอ้างหมายเลข Slave ID ด้วย ซึ่งก็หมายถึงว่า เรายังค่า Slave ID อยู่แล้วไม่จำเป็นต้องสั่งอ่านให้เสียเวลา แต่อาจจะใช้ประโยชน์คำสั่งนี้ในการตรวจสอบความพร้อมการทำงานของบอร์ดในระบบบัส ว่าบอร์ดหมายเลขอ้างกล่าวยังทำงานอยู่หรือไม่ ดังตัวอย่าง

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:3,0<CR>	Master สั่งอ่าน Holding Register[0] จาก Slave1
Slave ตอบรับ	*1:3,0=1<CR>	Holding Register[0] หรือ Slave ID = 1

ในกรณีที่ไม่ทราบค่าหมายเลข Slave ID ของบอร์ด ผู้ใช้สามารถใช้คำสั่งอ่านค่า Holding[10] ซึ่งเป็นรีสเตรอร์สำหรับบรรจุค่า Slave ID ของบอร์ด โดยอ้างหมายเลข Slave ID ในคำสั่งเป็น 0(Broadcast ID) ได้แต่กรณีนี้ต้องตอบบอร์ดในระบบบัสเพียง 1 ชุดเท่านั้น ไม่เช่นนั้นแล้วทุกบอร์ดในบัสจะตอบรับคำสั่งออกมากร้อนๆกันในบัสและทำให้ข้อมูลชนกัน ทำให้การสื่อสารผิดพลาดได้ โดยใช้รูปแบบคำสั่งดังตัวอย่าง

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*0:3,0<CR>	Master สั่งอ่าน Holding Register[9] จาก Slave ทุกด้วยในบัส
Slave ตอบรับ	*1:3,0=1<CR>	Holding Register[0] หรือ Slave ID = 1

2 การอ่านค่า Input Register

การสั่งอ่านค่า Input Register เป็นการสั่งค่าสถานะของ Sensor ต่างๆที่กำหนดไว้ ว่ามีค่าเป็นเท่าไร โดยจะใช้ Function 4 ในการอ่าน มีรูปแบบคำสั่งการใช้งานดังนี้

[*][ID][:][FUNC][,][ADDR0][<CR>]

- ID เป็นค่าหมายเลข Slave ID เป็นเลขจำนวนเต็มมีค่าระหว่าง 1 ถึง 247
- FUNC เป็นค่า Function ของคำสั่งมีค่า = 4(Read Input Register)
- ADDR0 ใช้กำหนดตำแหน่งรีลิสเตอร์สำหรับควบคุมการทำงานของ Relay
 - 0 หมายถึง Firmware Version
 - 1 หมายถึง Sensor Temperature Value
 - 2 หมายถึง Sensor Humidity Value

คำสั่งนี้จะให้รหัสค่า Function FUNC = 4 และแทนค่า ADDR0 ด้วยตัวเลข 0 ถึง 2 เพื่ออ้างถึงตำแหน่งของรีลิสเตอร์ที่ต้องการอ่านค่า ดังตาราง

Input Register(16 Bit Read Only Register)		
Address	Function	Data Type
0000	Read: Firmware Version (100 : V1.00)	ASCII unsigned (100 : 1.00)
0001	Read: Temperature Sensor (-40.0 ถึง +125.0)	ASCII Float (-40 ถึง 125.0 °C)
0002	Read: Humidity Sensor (0.0 ถึง 100.0)	ASCII Float (0.0 ถึง 100.0 %RH)

ตัวอย่างคำสั่ง

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:4,0<CR>	Master สั่งอ่าน Input Register[0] จาก Slave หมายเลข 1
Slave ตอบรับ	*1:4,0=100<CR>	ค่า Firmware Version = 1.00

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:4,1<CR>	Master สั่งอ่าน Input Register[1] จาก Slave หมายเลข 1
Slave ตอบรับ	*1:4,1=25.3<CR>	ค่า Temperature Sensor = 25.3 °C

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:4,2<CR>	Master สั่งอ่าน Input Register[2] จาก Slave หมายเลข 1
Slave ตอบรับ	*1:4,2=80.5<CR>	ค่า Humidity Sensor = 80.5 %RH

3 การกำหนดค่า Holding Register

การสั่งกำหนดค่า Holding Register เป็นการสั่งเขียนค่าเพื่อกำหนดค่า Slave ID หรือ กำหนดค่าชุดเซย์ค่าความผิดพลาดของอุณหภูมิ(Temperature Adjust) หรือ กำหนดค่าชุดเซย์ความผิดพลาดของความชื้น(Humidity Adjust) โดยจะใช้ Function 6 ในการเขียนข้อมูลให้รีจิสเตอร์ มีรูปแบบคำสั่งการใช้งานดังนี้

[*][ID][:][FUNC][,][ADDR0][=][VALUE0][<CR>]

- **ID** เป็นค่าหมายเลข **Slave ID** เป็นเลขจำนวนเต็มมีค่าระหว่าง 1 ถึง 247
- **FUNC** เป็นค่า Function ของคำสั่งมีค่า = 6(Write Holding Register)
- **ADDR0** ใช้กำหนดค่าແທນรีจิสเตอร์ที่ต้องการเขียนค่า
 - 0 หมายถึง กำหนดค่า Slave ID(ต้องกำหนด Jumper SET/RUN=SET ด้วย)
 - 1 หมายถึง กำหนดค่า Temperature Adjust
 - 2 หมายถึง กำหนดค่า Humidity Adjust
- **VALUE0** ใช้กำหนดค่าให้กับรีจิสเตอร์ โดยถ้า รีจิสเตอร์ตำแหน่งที่ 0 จะกำหนดค่าเป็นเลขจำนวนเต็มระหว่าง 1...247 แต่ถ้ารีจิสเตอร์ตำแหน่งที่ 1 จะกำหนดค่าเป็นเลขทศนิยมระหว่าง 0.0 ถึง +/-125.0 และถ้าเป็นรีจิสเตอร์ตำแหน่ง2 จะกำหนดเป็นค่าเลขทศนิยมระหว่าง 0.0 ถึง +/-100.0

ตัวอย่างคำสั่ง

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:6,0=2<CR>	Master สั่ง Slave1 ให้กำหนดค่า Slave ID = 2
Slave ตอบรับ	*2:6,0=OK<CR>	การทำงานถูกต้อง

- หมายเหตุ คำสั่งกำหนดค่า Slave ID ต้องกำหนด Jumper(SET/RUN) = SET ด้วยเสมอ
- การตอบรับคำสั่งจาก Slave จะตอบรับด้วย Slave ID ใหม่ที่ได้รับจากคำสั่ง

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:6,1=0.1<CR>	Master สั่ง Slave1 ให้กำหนดค่า Temperature Adjust = 0.1 °C
Slave ตอบรับ	*1:6,1=OK<CR>	การทำงานถูกต้อง

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*1:6,2=0.5<CR>	Master สั่ง Slave1 ให้กำหนดค่า Humidity Adjust = 0.5 %RH
Slave ตอบรับ	*1:6,2=OK<CR>	การทำงานถูกต้อง

การสั่งกำหนด Slave ID แบบ Broadcast(ADDR=0)

การรับ/ส่งคำสั่ง	รูปแบบคำสั่ง	รายละเอียดและความหมาย
Master ส่ง	*0:6,0=1<CR>	Master สั่ง Slave ที่ต้องอยู่ในบัส ให้กำหนดค่า Slave ID = 1
Slave ตอบรับ	*1:6,0=OK<CR>	การทำงานถูกต้อง

- หมายเหตุ คำสั่งกำหนดค่า Slave ID ต้องกำหนด Jumper(SET/RUN) = SET ด้วยเสมอ
- การตอบรับคำสั่งจาก Slave จะตอบรับด้วย Slave ID ใหม่ที่ได้รับจากคำสั่ง
- คำสั่งนี้ควรต่อ埠ดไว้ในบัสเพียง 1 ชุด หรือ ในกรณีที่ต้อง埠ดร่วมกันในบัสมากกว่า 1 ชุด ต้องกำหนด Jumper(SET/RUN) ของ埠ด ที่ต้องการกำหนดค่า Slave ID ในบัส ให้เป็น SET เพียง埠ดเดียว ไม่เช่นนั้นแล้ว埠ดทุก埠จะถูกกำหนดค่า Slave ID เหมือนกันทั้งหมด