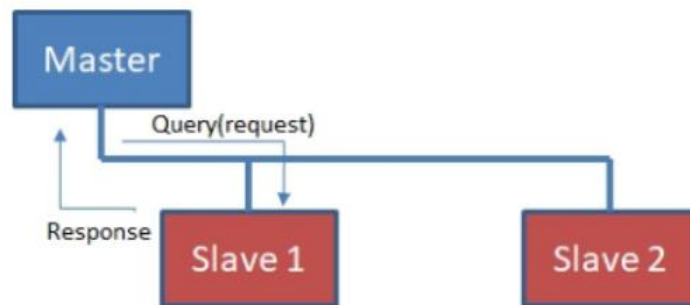


การสื่อสารผ่าน Modbus Protocol

Modbus คือ โพรโตคอล (Protocol) การสื่อสารที่พัฒนาขึ้นโดย บริษัท Modicon Systems ด้วยรูปแบบง่ายๆ เป็นรูปแบบการส่งข้อมูลระหว่างอุปกรณ์อิเล็กทรอนิกส์ อุปกรณ์ที่ต้องการข้อมูลเรียกว่า Modbus Master (Client) ส่วนอุปกรณ์ที่ให้ข้อมูลที่ต้องการเรียกว่า Modbus Slave (Server) ใน Modbus Network ที่เป็นมาตรฐานนั้นจะมี Master ตัวเดียวแต่ Slave มีได้ถึง 247 ตัว โดยแต่ละตัวจะมี ID ระบุเหมือนเลขที่บ้านตั้งแต่ 1 ถึง 247 และ Master สามารถ Write ข้อมูลไปยัง Slave ได้

Modbus เป็น Open Protocol หมายความว่า บุคคลทั่วไปสามารถพัฒนาอุปกรณ์ที่ใช้การสื่อสารแบบ Modbus โดยไม่ต้องเสียค่าใช้จ่ายใดๆ Modbus จึงเป็น Protocol พื้นฐานและนิยมใช้อย่างแพร่หลายในทุกอุตสาหกรรม โดยใช้รับส่งข้อมูลจากอุปกรณ์ควบคุมกับ Controller หรือระบบประมวลผลข้อมูลต่าง

Slave อาจเป็นอุปกรณ์ต่อพ่วงใดๆ เช่น Input/Output Transducer, วาล์ว (Valve), Inverter, อุปกรณ์บันทึกข้อมูล (Data Logger) หรืออุปกรณ์เครื่องมือวัดอื่นๆ เป็นต้น ซึ่งประมวลผลและส่งข้อมูลไปยัง Master



แสดงการสื่อสารระหว่าง Master กับ Slave

Master สามารถติดต่อกับ Slave แต่ละตัวได้หรือสามารถส่งเป็น Message ถึง Slave ทุกตัวได้ในลักษณะของการ Broadcast และ Slave จะตอบสนองสิ่งที่ Master ต้องการเท่านั้น สิ่งที่ Master ส่งให้จะประกอบด้วย Slave Address, Function Code (คำสั่งหรือสิ่งที่ต้องการให้ทำ), Data และ Checksum ส่วนข้อมูลที่ Slave ส่งกลับมาจะประกอบด้วยคำสั่งที่สั่งให้กระทำหรือข้อมูลต่างๆ และ Checksum

Modbus แบ่งออกเป็น Modbus Serial: ASCII/RTU (เป็นพอร์ตการสื่อสารแบบอนุกรม RS232, RS485, RS422) และ Modbus TCP/IP (LAN) เพื่อให้สอดคล้องกับแนวโน้มการพัฒนาการสื่อสารในปัจจุบันและทุกอย่างที่สามารถเชื่อมต่อกับเครือข่าย Ethernet หรือ Internet เพื่อส่งข้อมูล

- ❖ **Modbus Serial** เป็นการสื่อสารโดยการส่งข้อมูลไปตามสายสัญญาณ Serial ระหว่างอุปกรณ์ โดยวิธีการสื่อสารที่ง่ายที่สุดคือการต่อสายสัญญาณ Serial ระหว่าง Master หนึ่งตัวกับ Slave หนึ่งตัว (Point-to-Point) ซึ่งจะเป็นการสื่อสารผ่านพอร์ต RS232 หรือ RS422 หรือการต่อ Master หนึ่งตัวกับ Slave หลายตัว (Point-to-Multipoint) ซึ่งจะเป็นการสื่อสารผ่านพอร์ต RS485

การรับส่งข้อมูลด้วยโปรโตคอล Modbus Serial สามารถเลือกได้ 2 โหมด คือ โหมด ASCII และ โหมด RTU ซึ่งทั้ง 2 โหมดนี้มีความแตกต่างกันที่การกำหนดรูปแบบของชุดข้อมูลภายในเฟรม จะเลือกโหมดใดก็ได้แต่มีเงื่อนไขว่าอุปกรณ์ทุกตัวที่ต่อร่วมกันอยู่ในบัสหรือเครือข่ายเดียวกัน จะต้องตั้งให้เลือกใช้โหมดเดียวกันทั้งหมด รวมถึง Serial Parameter ต่างๆ เช่น Baud Rate, Data Bit, Stop Bit และ Parity Bit

❖ **Modbus TCP/IP** ถูกพัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อนำการสื่อสารแบบ Ethernet มาใช้กับอุปกรณ์จำพวก Ethernet Device ระยะในการใช้งานสำหรับการเดินสาย (สาย LAN) คือ 100 เมตร โดยสามารถขยายระยะในการสื่อสารได้โดยการใช้อุปกรณ์ Repeater หรือในระบบ LAN จะเรียกอุปกรณ์นี้ว่า Hub หรือ Switch ก็จะสามารถลากสายได้อีก 100 เมตร และยังสามารถต่อ Repeater ขยายระยะทางได้โดยไม่จำกัด ในการสื่อสารโดยทั่วไปมีความเร็ว 100,000,000 บิตต่อวินาที (100 Mbps) และเชื่อมต่ออุปกรณ์ได้ไม่จำกัดจำนวน

สำหรับอุปกรณ์ Modbus Serial ที่จะติดต่อสื่อสารกับอุปกรณ์ Modbus TCP/IP เพื่อให้ใช้งานในเครือข่าย Ethernet จะใช้ Gateway ติดต่อและแปลงรูปแบบการสื่อสารข้อมูล โดยการสื่อสารของ Modbus Serial จะถูก Gateway แปลงให้เป็น Modbus TCP/IP เพื่อใช้ในการติดต่อสื่อสารในเครือข่าย Ethernet

Modbus Function Codes

การร้องขอข้อมูลของ Master จะต้องระบุถึง Function Code สำหรับการเข้าถึง Modbus Register ที่ Slave ร้องรับให้ถูกต้อง ถ้าระบุไม่ตรงกับที่ Slave กำหนดไว้จะทำให้ Slave ไม่ส่งค่ากลับไปยัง Master โดย Modbus Function Codes มีรายละเอียดดังนี้

Function Code	Register Type	Descriptions
01	Read Coil	แสดงสถานะของ Digital Output (0 = OFF, 1 = ON)
02	Read Discrete Input	แสดงสถานะของ Digital Input (0 = OFF, 1 = ON)
03	Read Holding Registers	แสดงค่าของ Analog Output
04	Read Input Registers	แสดงค่าของ Analog Input
05	Write Single Coil	เขียนค่ากลับไปยัง Digital Output (เขียนค่าได้ที่ละ 1 Register)
06	Write Single Holding Register	เขียนค่ากลับไปยัง Analog Output (เขียนค่าได้ที่ละ 1 Register)
15	Write Multiple Coils	เขียนค่ากลับไปยัง Digital Output (เขียนค่าได้ที่ละหลาย Register)
16	Write Multiple Holding Registers	เขียนค่ากลับไปยัง Analog Output (เขียนค่าได้ที่ละหลาย Register)

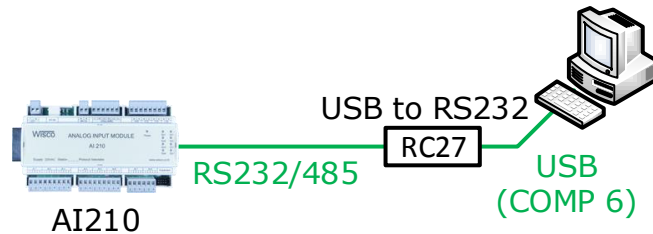
ตัวอย่างการอ่านค่าจากอุปกรณ์ Modbus

โดยตัวอย่างนี้จะใช้อุปกรณ์ของทาง บริษัท Wisco รุ่น AI210 (Analog Input Module) ที่มีพอร์ต RS232/485 สื่อสารด้วย Protocol Modbus RTU และใช้โปรแกรม MODSCAN สำหรับอ่านค่า ซึ่งจะแสดงการเชื่อมต่อ 2 แบบ คือ การเชื่อมต่อผ่านทาง Serial และการเชื่อมต่อผ่านทาง Ethernet โดยอาศัยตัวแปลงสัญญาณจาก Modbus Serial เป็น Modbus TCP/IP มีรายละเอียดดังนี้

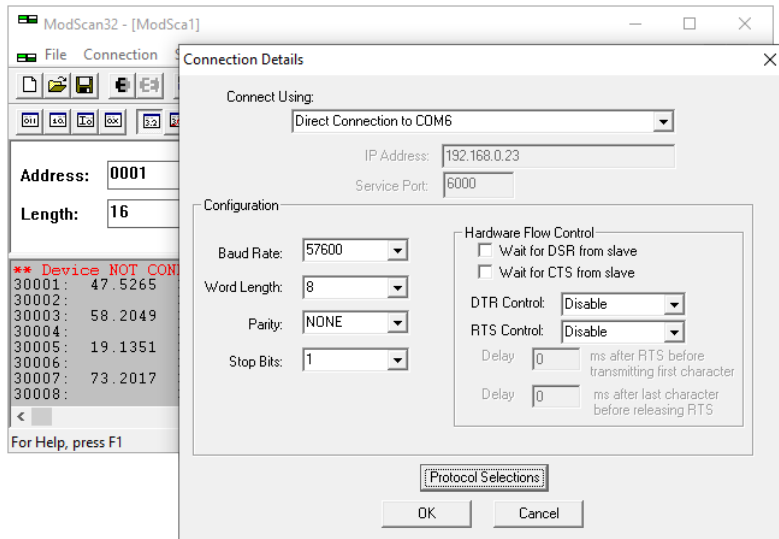


ข้อมูลของ **AI210**

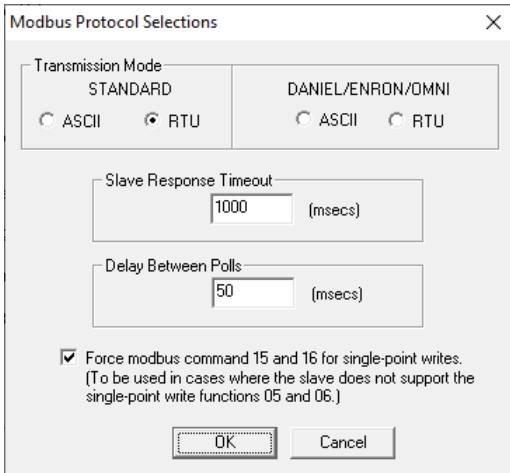
❖ การเชื่อมต่อผ่านทาง **Modbus Serial (RS232/485)**



จากรูป จะเป็นการเชื่อมต่อ AI210 ผ่านตัวแปลงสัญญาณจาก RS232/485 เป็น USB เข้ากับเครื่องคอมพิวเตอร์ (ในกรณีที่เครื่องคอมพิวเตอร์ไม่มีพอร์ต RS232)



- 1) เปิดโปรแกรม Modscan ขึ้นมา จากนั้นเลือกเมนู Connection -> Connect
- 2) ที่หน้าต่าง "Connection Details" เลือก Serial Port ที่ใช้งานอยู่ (จากตัวอย่างคือ COMP 6) และตั้งค่า "Configuration" เช่น Baud Rate, Word Length (Data Bit), Parity และ Stop Bit ให้ตรงกับ AI210 (หรืออุปกรณ์ Modbus อื่นๆที่เชื่อมต่อ)

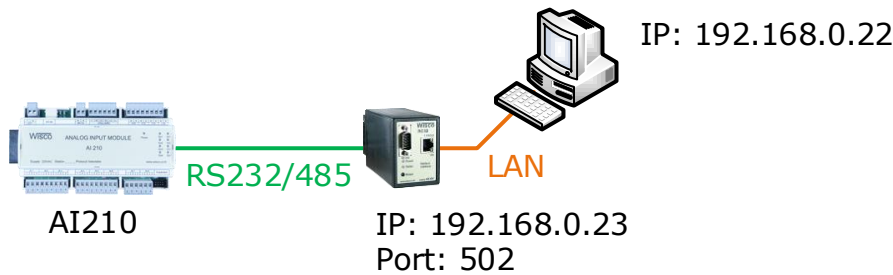


Slave Response Timeout กำหนดเวลารอ Slave ส่งข้อมูลกลับมา ถ้า Slave ไม่ส่งข้อมูลกลับมาตามเวลาที่กำหนด Master จะทำการส่งคำสั่งชุดใหม่ออกไป

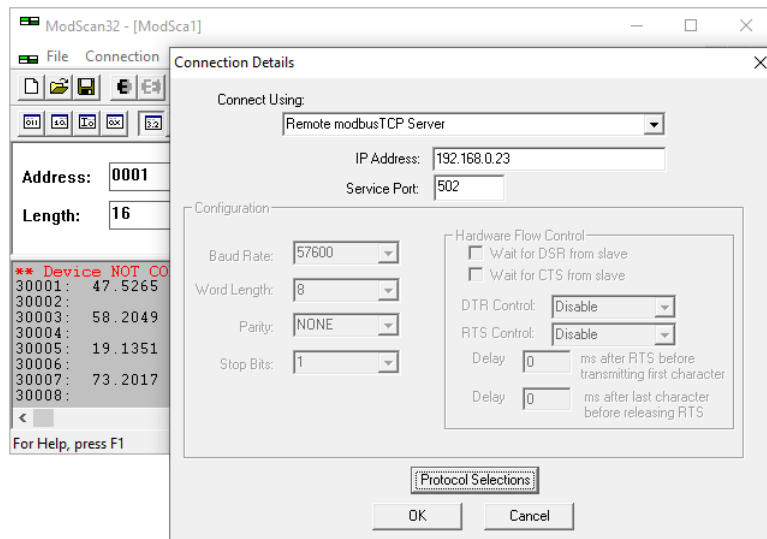
Delay Between Polls กำหนดเวลาสำหรับรอส่งคำสั่งชุดต่อไปหลังจากได้รับข้อมูลเข้ามาแล้ว

3) จากนั้นกดปุ่ม "Protocol Selections" ที่หัวข้อ "Transmission Mode: Standard" กำหนดเป็น RTU หลังจากนั้นกดปุ่ม "OK" เพื่อทำการเชื่อมต่อกับ AI210

❖ การเชื่อมต่อผ่านทาง Modbus TCP/IP

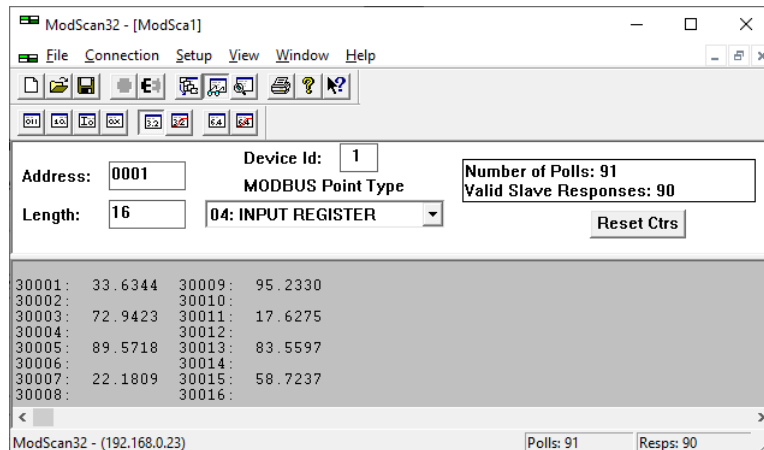


จากรูป จะเป็นการเชื่อมต่อ AI210 ผ่านตัวแปลงสัญญาณจาก RS232/485 เป็น Ethernet (Modbus Gateway: RC32) เพื่อแปลงสัญญาณจาก Modbus Serial เป็น Modbus TCP/IP



- 1) เปิดโปรแกรม Modscan ขึ้นมา จากนั้นเลือกเมนู Connection -> Connect
- 2) ที่หน้าต่าง "Connection Details" ที่หัวข้อ "Connect Using" เลือกเป็น Remote Modbus TCP Server และกำหนด IP กับ Port ของ Gateway (ในตัวอย่างนี้จะระบุ IP และ Port ที่ RC32 เปิดรอไว้) หลังจากนั้นกดปุ่ม "OK" เพื่อทำการเชื่อมต่อกับ AI210

❖ การอ่านค่าจาก AI210



หลังจากทำการเชื่อมต่อแล้วให้กำหนดค่าของ Register Address ตามที่อุปกรณ์ AI210 กำหนดไว้ ดังนี้

- 1) ที่หัวข้อ "Modbus Point Type" กำหนดเป็น 04: Input Register
- 2) ที่เมนู Setup -> Display Options เลือก Floating Pt
- 3) ที่หัวข้อ "Address" กำหนด Address เริ่มต้นที่ต้องการจะอ่านค่า เช่น 1 (Channel 1: 40001) และกำหนดจำนวน Address ที่ต้องการจะอ่านทั้งหมด (16 Address หมายถึง อ่านค่าจาก AI210 เริ่มจาก Channel 1 - 8, FLOAT ใช้ 2 Address ในการอ่านค่า)

อ้างอิง: <https://medium.com/maestro19>
<https://medium.com/@aegkaluk>