

# EM-TNET

## V2.0 (2020)

### Temperature (High) to RS485

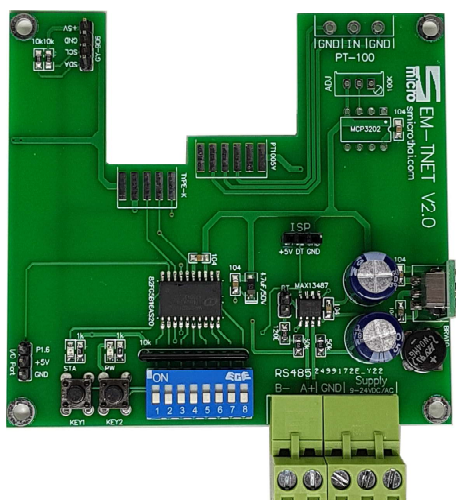


EM-TNET คือบอร์ดวัดอุณหภูมิ ใช้ Sensor มาตรฐานได้ถึง 3 รุ่น คือ PT100 , Type-K และ GY-906 (แบบอินฟาเรด ไม่ต้องสัมผัสวัตถุ) ตัวบอร์ดไม่มีการแสดงค่าอุณหภูมิ แต่สามารถดูค่าได้ผ่านการสื่อสาร RS485 หรือจะใช้ต่อกับตัวแสดงผลระยะไกลก็ได้ มี I/O Digital 1 Bit ที่ควบคุมผ่านทาง RS485 ได้ด้วยการตั้งค่าต่าง ๆ ทำได้ง่ายด้วย Dip-Switch 8 ตัว นำประยุกต์ใช้งานต่าง ๆ ได้คุ้มค่าสูงสุด

#### คุณสมบัติ

- ทำงานด้วยไมโครคอนโทรลเลอร์เบอร์ 82FG5B16 ความถี่ภายใน 11.0592 MHz
- มี LED สีแดงแสดง Power และ LED เหลืองแสดงสถานะต่าง ๆ
- มีขั้ว 3 Pin เป็น I/O Digital อีก 3 Bit ประยุกต์ใช้งานต่าง ๆ ได้ตามต้องการ
- มีพอร์ท RS485 เพื่อการดูค่าอุณหภูมิ รวมทั้งควบคุม I/O Digital ผ่านการสื่อสารจากคอมพิวเตอร์ได้ รวมทั้งสามารถต่อพ่วงกันหลาย ๆ บอร์ดบนสาย RS485 เดียวกันได้ และเลือกสื่อสารกับแต่ละบอร์ดด้วยการตั้ง Address แตกต่างกันไปได้ถึง 16 ตัว
- คุณสมบัติการสื่อสาร RS485 คือ Baud-Rate=9600 Data=8 Stop=1 Parity=No
- เลือกใช้ชุดคำสั่งแบบ Sac (Smart Ascii Command) หรือ ModBus ก็ได้ หรือเลือกใช้โหมด Steam เพื่อส่งค่า Temp ออกทุก ๆ วินาทีก็ได้ (ใช้แสดงตัวเลขระยะไกล และขนาดใหญ่ได้)
- โหมด Steam สามารถตั้ง Calibrate ได้ด้วย โดยมีตัวเลขตั้งแต่ -7 ถึง 7 องศาเซลเซียส
- เลือกใช้กับหัว Sensor ได้ 3 รุ่น (ความละเอียด 1 องศา) ดังนี้  
 PT100 ... วัด -100 ถึง 400 องศาเซลเซียส (บนบอร์ดต้องเพิ่มโมดูล PT1005V และชิพ MCP3202)  
 Type-K ... วัด 0 ถึง 800 องศาเซลเซียส (บนบอร์ดต้องเพิ่มโมดูล TYPE-K)  
 GY-906 ... วัด -70 ถึง 380 องศาเซลเซียส (เป็นแบบอินฟาเรด ไม่ต้องสัมผัสวัตถุ ระยะ 50 cm)
- มีช่อง VR 100K เพื่อการทดสอบ โดยปรับได้ 0-999(1000) ระดับ (บนบอร์ดต้องเพิ่มชิพ MCP3202)
- ใช้ไฟเลี้ยง 9 VAC หรือ 12 VDC กินกระแส 20 mA
- ขนาดบอร์ด 90 x 85 mm (ไม่รวมหัววัด Sensor)

#### ภาพแสดงบอร์ดและ Option



## การตั้งคุณสมบัติด้วย Dip-Switch

EM-TNET ใช้การตั้งคุณสมบัติทั้งหมดด้วย Dip-Switch ซึ่งง่ายและสะดวก และไม่ต้องใช้อุปกรณ์อื่นใดเพิ่มเติม โดยครอบคลุมการตั้งรุ่น Sensor, โหมดในการใช้งาน และ Address ของบอร์ด ดังนี้

### รุ่นของ Sensor

Dip1=OFF Dip2=OFF	โหมดทดสอบ โดยปรับค่าที่ VR 100K (บนบอร์ดต้องเพิ่มชิพ MCP3202) โดยจะส่งค่าตั้งแต่ 0-999 (ภายในจะทำงานจริงที่ 0-1000)
Dip1=OFF Dip2=ON	ใช้กับหัวรุ่น PT100 (บนบอร์ดต้องเพิ่มโมดูล PT1005V และชิพ MCP3202)
Dip1=ON Dip2=OFF	ใช้กับหัวรุ่น Type-K (บนบอร์ดต้องเพิ่มโมดูล TYPE-K)
Dip1=ON Dip2=ON	ใช้กับหัวรุ่น GY-906 ต่อเข้ากับขั้ว 4 Pin

### โหมดการใช้งาน (การสื่อสารทาง RS485)

Dip3=OFF Dip4=OFF	สื่อสารแบบ SAC (Smart Ascii Command)
Dip3=OFF Dip4=ON	สื่อสารแบบ ModBus
Dip3=ON Dip4=OFF	ส่งข้อมูลแบบ Steam ทุก ๆ วินาที
Dip3=ON Dip4=ON	ส่งข้อมูลแบบ Steam-Display ทุก ๆ วินาที

### ตั้ง Address (BCD) ของบอร์ด (สำหรับโหมด SAC และ ModBus)

Dip5=OFF Dip6=OFF Dip7=OFF Dip8=OFF	Address 0x00
Dip5=OFF Dip6=OFF Dip7=OFF Dip8=ON	Address 0x01
Dip5=OFF Dip6=OFF Dip7=ON Dip8=OFF	Address 0x02
Dip5=OFF Dip6=OFF Dip7=ON Dip8=ON	Address 0x03
Dip5=OFF Dip6=ON Dip7=OFF Dip8=OFF	Address 0x04
Dip5=OFF Dip6=ON Dip7=OFF Dip8=ON	Address 0x05
Dip5=OFF Dip6=ON Dip7=ON Dip8=OFF	Address 0x06
Dip5=OFF Dip6=ON Dip7=ON Dip8=ON	Address 0x07
Dip5=ON Dip6=OFF Dip7=OFF Dip8=OFF	Address 0x08
Dip5=ON Dip6=OFF Dip7=OFF Dip8=ON	Address 0x09
Dip5=ON Dip6=OFF Dip7=ON Dip8=OFF	Address 0x10
Dip5=ON Dip6=OFF Dip7=ON Dip8=ON	Address 0x11
Dip5=ON Dip6=ON Dip7=OFF Dip8=OFF	Address 0x12
Dip5=ON Dip6=ON Dip7=OFF Dip8=ON	Address 0x13
Dip5=ON Dip6=ON Dip7=ON Dip8=OFF	Address 0x14
Dip5=ON Dip6=ON Dip7=ON Dip8=ON	Address 0x15

## ตั้งค่า Calibrate (สำหรับโหมด Steam)

Dip5=OFF Dip6=OFF Dip7=OFF Dip8=OFF	Temp +0
Dip5=OFF Dip6=OFF Dip7=OFF Dip8=ON	Temp +1
Dip5=OFF Dip6=OFF Dip7=ON Dip8=OFF	Temp +2
Dip5=OFF Dip6=OFF Dip7=ON Dip8=ON	Temp +3
Dip5=OFF Dip6=ON Dip7=OFF Dip8=OFF	Temp +4
Dip5=OFF Dip6=ON Dip7=OFF Dip8=ON	Temp +5
Dip5=OFF Dip6=ON Dip7=ON Dip8=OFF	Temp +6
Dip5=OFF Dip6=ON Dip7=ON Dip8=ON	Temp +7

Dip5=ON Dip6=OFF Dip7=OFF Dip8=OFF	Temp -0
Dip5=ON Dip6=OFF Dip7=OFF Dip8=ON	Temp -1
Dip5=ON Dip6=OFF Dip7=ON Dip8=OFF	Temp -2
Dip5=ON Dip6=OFF Dip7=ON Dip8=ON	Temp -3
Dip5=ON Dip6=ON Dip7=OFF Dip8=OFF	Temp -4
Dip5=ON Dip6=ON Dip7=OFF Dip8=ON	Temp -5
Dip5=ON Dip6=ON Dip7=ON Dip8=OFF	Temp -6
Dip5=ON Dip6=ON Dip7=ON Dip8=ON	Temp -7

## การใช้งานทั่วไป และโหมด Steam

เมื่อจ่ายไฟเข้าบอร์ด LED สีเหลืองจะกระพริบ 2 ครั้ง จากนั้นจะสว่างครู่หนึ่งแล้วจึงดับไป แสดงว่าพร้อมสำหรับการใช้งานแล้ว  
ถ้ามีคำสั่งเข้ามาทาง RS485 โดยต้องเป็นคำสั่งที่ถูกต้องและมี Address ตรงกับบอร์ดที่ตั้งไว้ (ถ้าเลือกใช้งานแบบมี Address)  
ตัว LED สีเหลืองก็จะกระพริบตามจังหวะสื่อสารให้ทราบด้วย กรณี LED สีเหลืองสว่าง "ค้าง" จะมีความหมายหลายประการดังนี้

# รอเวลาให้ค่าที่อ่านจาก Sensor มีความเสถียรก่อน ซึ่งโดยปกติจะใช้เวลาประมาณ 2-3 วินาที

# Sensor มีปัญหา ซึ่งอาจจะเกิดจากสายขาด หรือตัว Sensor เสียหาย ยกเว้นรุ่น PT100 ซึ่งจะไม่สามารถทราบได้  
(ทดสอบได้ด้วย การตั้ง Sensor ออก ตัว LED ก็จะสว่างค้าง)

ในโหมด Steam นั้น เครื่องจะส่งข้อมูลออกจาก RS485 ทุก ๆ วินาทีดังนี้

:Ttt\r ttt คือค่าอุณหภูมิ 3 หลัก เป็นองศาเซลเซียส ความละเอียด 1 องศา  
โดยจะแสดงได้สูงสุดตามคุณสมบัติของ Sensor ส่วนค่าติดลบจะแสดง  
ได้ต่ำสุดที่ -99 องศาเซลเซียส

ส่วนในโหมด Steam-Display ก็จะทำนองเดียวกัน เพียงแต่จะเปลี่ยนอักษร T เป็นหมายเลข 1 เท่านั้น ซึ่งจะสอดคล้องกับสินค้า  
Display ของเอสไมโคร โดยสามารถนำไปแสดงผลด้วยตัวเลขในระยະไกล หรือตัวเลขใหญ่ได้ตามต้องการ กรณี LED สว่างค้างอยู่  
คือ Sensor มีปัญหาหรือรอเสถียร ค่าที่ส่งจะเป็น :T—\r ให้รับทราบด้วย

## ชุดคำสั่ง RS485 แบบ SAC

คุณสมบัติพื้นฐาน RS485 คือ Baud-Rate=9600 Parity=None Data=8 Stop=1 สำหรับชุดคำสั่งแบบ Sac Protocol คือ Smart Ascii Command สามารถใช้โปรแกรมสื่อสารแบบ Terminal ที่ทำงานบนเครื่องคอมพิวเตอร์ เพื่อการทดสอบคำสั่งได้ทันที เพราะมีเป็นตัวอักษร Ascii ทั้งหมด ชุดคำสั่งจะมีรูปแบบดังนี้...

:c\r สำหรับใช้งานแบบไม่มี Node Address คือใช้งานเพียงบอร์ดเดียว

:@aac\r สำหรับใช้งานแบบมี Node Address คือใช้พ่วงกันหลาย ๆ บอร์ด

: คือรหัสนำหน้า (0x3A)

c คือรหัสคำสั่งเป็นตัวเลข หรือตัวอักษรตัวเดียว

\r คือรหัสลงท้าย (0x0D)

@ คือรหัสเพิ่มเติม (0x40) กรณีใช้งานแบบมี Node Address

aa คือหมายเลข Node Address ซึ่งเป็น Ascii แบบ bcd 01-99

ตัวอย่างเช่นถ้าหมายเลข aa=37 รหัส Ascii ก็คือ 0x33 และ 0x37

(ความจริง aa รองรับเลข Hexฐานสิบหก แต่แนะนำให้ใช้เป็น bcd เพื่อให้ดูเข้าใจได้ง่าย)

Sac เป็น Protocol ที่ทำให้ใช้งานแบบตัวเดียวก็ได้ หรือพ่วงกันเป็น Network ก็ได้ โดยทำการตั้ง Node Address ได้จากชุดคำสั่ง โดยถ้าเป็นการใช้งานบอร์ดเดียว ก็สามารถใช้รูปแบบ :c\r ใช้งานได้เลย เมื่อส่งคำสั่งไปแล้ว บอร์ดจะตอบสนองกลับมาด้วยข้อมูลหรือข้อความใด ๆ เสมอ และถ้าใช้พ่วงกันหลาย ๆ บอร์ด ก็ให้ตั้ง Node Address แต่ละบอร์ดแตกต่างกันไป และก็ใช้คำสั่งรูปแบบ :@aac\r ซึ่งเฉพาะบอร์ดที่มี Node Address ตรงกับ aa เท่านั้นที่จะตอบสนองกลับ

ชุดคำสั่งจะสรุปได้ตามตารางต่อไปนี้ โดยจะแสดงในรูปแบบไม่มี Node Address และไม่ใส่รหัสลงท้าย \r เพื่อให้ดูสบายตา ส่วนคำภายใน [ ] หมายถึง Option คือเลือกใส่หรือไม่ใส่ก็ได้ ซึ่งความหมายส่วนใหญ่ก็คือ การใส่ Option คือการ “เขียนค่า” (write) และการไม่ใส่คือการ “อ่านค่า” (read) ในวงเล็บ (R) หมายถึง รับข้อมูลจากตัวแม่ และ (T) หมายถึง ข้อมูลที่ส่งกลับไปยังตัวแม่

---

(R) :1

(T) Txxx

Read Temp ... สำหรับอ่านค่าอุณหภูมิ ณ ขณะนั้น โดย xxx คือค่าอุณหภูมิ ซึ่งแสดงค่าติดลบได้ถึง -99 ด้วย กรณีถ้าแสดงเป็น — แสดงว่าอยู่ในช่วงรอเสถียร หรือ Sensor มีปัญหา

---

(R) :2[x]

(T) x[OK]

Read/Write I/O-Port ... สำหรับการอ่านหรือเขียนค่าให้กับ I/O-Port ที่ขั้ว 3 Pin โดย x คือสถานะของ Port โดยถ้าเป็น 0 หมายถึง I/O=5V และถ้าเป็น 1 หมายถึง I/O=GND (Active Low)

---

(R) :X[aa]

(T) aa[OK]

Read/Write Node Address ... โดย aa คือค่า Address 01-99 (bcd หรือ Hex)

---

(R) :Y[aaaa]

(T) ข้อความต่าง ๆ

Checksum or Dump Flash ... สำหรับแสดงค่า Size และ Checksum ของ Firmware เพื่อตรวจสอบความถูกต้อง และถ้าใส่ aaaa คือเลข hex ของ address ก็จะแสดงข้อมูลในหน่วยความจำ Flash ภายในตัว MCU นั้นเอง (จะแสดงเฉพาะส่วน IAP) โดยแสดงเป็นจำนวน 128 Byte

---

(R) :Z[!]

(T) ข้อความต่าง ๆ

Show Model / Version & Self-Test ... สำหรับแสดงชื่อสินค้า และรุ่นของ Firmware เป็น vX.X และถ้าใส่ ! จะหมายถึงเข้าระบบ Self-Test เพื่อการตรวจสอบและเพื่อการซ่อม (ถ้ามี)

---

หมายเหตุ ... สินค้าของเอสไมโครที่มี Sac Protocol คำสั่ง X,Y,Z จะเป็นรูปแบบทำนองเดียวกันเสมอ

Sac Protocol ยังรองรับการทำงานที่ซับซ้อนขึ้น เพื่อให้เหมาะกับสภาพแวดล้อมที่แตกต่างกันไป โดยถ้าใส่อักษร & (0x26) ต่อท้ายคำสั่ง คืออยู่ท้ายสุดของคำสั่ง Ascii แต่อยู่ก่อนรหัสลงท้าย \r (0x0D) จะหมายถึงไม่ต้องการให้มีการตอบกลับใด ๆ โดยที่เครื่องยังคงทำงานตามคำสั่งทุกประการ ซึ่งมักจะใช้กับคำสั่งเพื่อการตั้งค่าหรือแสดงข้อมูล และไม่ต้องการให้ตอบกลับเพื่อความรวดเร็วในการทำงาน ... และสำหรับสภาพการสื่อสารที่มีสัญญาณรบกวนสูง อาจจะต้องการตรวจสอบคำสั่งให้มีความแน่นอนมากขึ้น ก็สามารถใส่ค่า Checksum เพิ่มเติมได้ตามตัวอย่างนี้

:Z@5A\r

EM-TNET v2.0 (2020)@55\r

ค่า Checksum คือผลบวก Hex ของอักขรทุกตัวหลังจาก : จนถึง @ ตัวอย่าง @5A คือค่า Checksum ของอักขร Z และ @55 คือค่า Checksum ของ EM-TNET v2.0 (2020) ทั้งนี้ถ้าค่า Checksum ไม่ถูกต้อง เครื่องก็จะไม่ทำงานใด ๆ สำหรับข้อมูลที่ตอบกลับก็เช่นกัน ถ้าตรวจสอบค่า Checksum แล้วไม่ถูกต้อง ก็ให้ถือว่าเป็นข้อมูลที่ไม่มีความหมายใด ๆ เพราะน่าจะมีความผิดพลาดระหว่างการสื่อสารนั่นเอง

## ชุดคำสั่ง RS485 แบบ ModBus

คุณสมบัติพื้นฐาน RS485 คือ Baud-Rate=9600 Parity=None Data=8 Stop=1 สำหรับชุดคำสั่งแบบ ModBus (RTU-Binary) ซึ่งเป็นมาตรฐานที่ใช้กันโดยทั่วไปในแวดวงอุตสาหกรรม ทั้งนี้ EM-TNET จะเป็นตัว Slave คือเมื่อมีคำสั่งเข้ามา ตัว EM-TNET ก็จะตอบกลับไปด้วยข้อมูล สำหรับโหมด ModBus นี้ จะรับคำสั่งแบบ Fix ตายตัว ไม่ได้ยืดหยุ่นให้ปรับตัวแปรต่างๆ ได้ต้องการที่จะมี Format เป็นตัวเลข Hex มาตรฐานดังนี้

รับคำสั่ง (R)      ## 04 00 00 00 01 XX XX

## คือหมายเลข Node-Address (Hex)

04 คือ Function Code (Read Analog Input Register)

00 00 คือตำแหน่งตัวแปรที่จะอ่าน (สำหรับ EM-TNET มีเพียงตำแหน่งเดียวเท่านั้น)

00 01 คือจำนวน Register ที่จะอ่าน (สำหรับ EM-TNET ต้องเป็น 00 01 เท่านั้น)

XX XX คือค่า CRC Checksum

ส่งข้อมูลกลับ (T)      ## 04 02 AA AA XX XX

## คือหมายเลข Node-Address (Hex)

04 คือ Function Code (Read Analog Input Register)

02 คือความยาวของข้อมูลที่จะตามมา

AA AA คือข้อมูล Data (ค่า Signed Integer 16 บิต)

XX XX คือค่า CRC Checksum

ตำแหน่งตัวแปร 00 00 (30001 Coil/Register Numbers) คืออ่านค่า Temp โดยเป็นตัวเลข Signed Integer รองรับค่า Temp ติดลบได้ด้วย เช่น ถ้าได้ค่าเป็น 0x0014 คือ Temp=20 องศา และถ้าได้ค่าเป็น 0xFFFF8 คือ Temp=-8 องศานั่นเอง