

# EM-D40A

## v2.2 (2020)

### 7-Segment LED Display



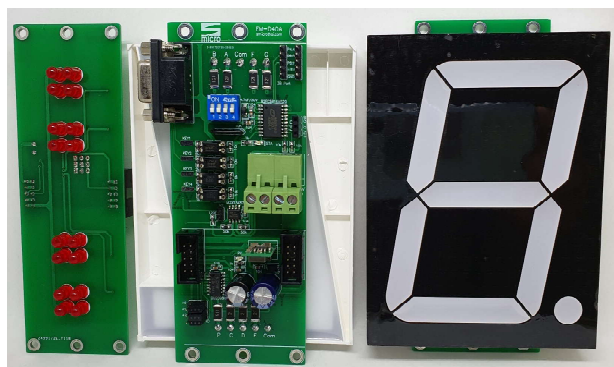
บอร์ดแสดงผล 7-Segment LED ที่รับคำสั่งควบคุมผ่านทาง RS485 หรือ 3B-Port ใช้ต่อกับคอมพิวเตอร์ไมโครต่าง ๆ เพื่อการแสดงผลได้อย่างสะดวก สามารถต่อพ่วงกันได้หลายชุดในแบบ RS485 Network ได้โดยการตั้งหมายเลข Address ให้แตกต่างกัน แต่ละชุดตัวเลข ยังสามารถเลือกต่อตัวเลขได้ตั้งแต่ 1 ถึง 8 หลัก โดยให้บอร์ดหลักร่วมกับบอร์ดขยาย ตัวสินค้ามีให้เลือกดังนี้

- EM-D40AM บอร์ดหลัก (Main) ตัวเลขสูง 4 นิ้ว มีชุด MCU ควบคุมการทำงาน
- EM-D40AD บอร์ดขยาย (Digit) ตัวเลขสูง 4 นิ้ว สำหรับต่อพ่วงตามต้องการ
- EM-D40AC บอร์ดขยาย (Colon) แสดง Colon หรือจุด หรือองศา

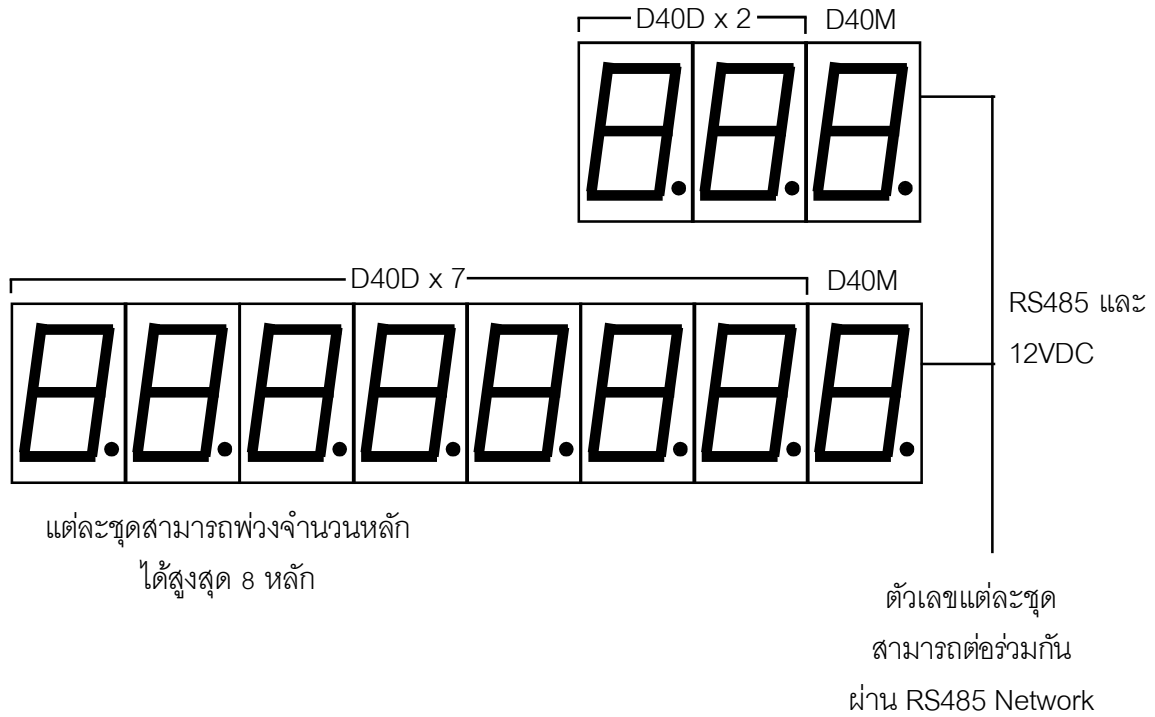
#### คุณสมบัติของบอร์ดหลัก

- ทำงานด้วยไมโครคอนโทรลเลอร์เบอร์ 82FG5B16 ความถี่ภายใน 11.0592 MHz
- มี LED สีแดงแสดง Power และสีเหลืองแสดงสถานะต่าง ๆ
- มีพอร์ต RS485 เพื่อรับคำสั่งแบบ Sac (Smart Ascii Command) เพื่อการแสดงผลต่าง ๆ
- คุณสมบัติการสื่อสาร RS485 คือ Baud-Rate=9600 Data=8 Stop=1 Parity=No
- มีขั้ว 5 Pin คือ 3B-Port ที่ขนานกัน 2 ขั้ว ใช้เพื่อเป็น I/O Digital ควบคุมผ่าน RS485 ได้ หรือใช้เป็น Input เพื่อการแสดงผลตัวเลข โดยใช้สัญญาณแบบ 595 Low Speed (DIN,CLK,STR)
- มีขั้ว 3 Pin คือ IO-Port ใช้เพื่อเป็น I/O Digital ควบคุมผ่าน RS485 ได้
- มี Dip-Switch เพื่อการตั้ง Address ของ RS485 Network
- มีจุดต่อ Input-Key 1-4 แบบ Active Low สำหรับการต่อกับปุ่มกด หรือกำหนดโหมดควบคุมตัวเลขผ่าน 3B-Port แบบ 595 Low Speed
- มีขั้ว 10 Pin สำหรับต่อตัวเลข (Digit) เพิ่มด้วยบอร์ดขยาย โดยใช้สายแพ 10 เส้น ซึ่งจะพ่วงไปครบถ้วน ทั้งด้านไฟเลี้ยง และสัญญาณต่าง ๆ
- ตัวเลขแต่ละหลัก ทั้งในบอร์ดหลักและบอร์ดขยาย มี Jumper 3 ตัวเพื่อเลือกตำแหน่ง Digit ได้อิสระ
- ไฟเลี้ยง 12VDC พร้อมกับ RS485 ผ่านทางขั้วเขียวไขว้ชนิด
- และมี DB9 ตัวเมีย รองรับไฟเลี้ยง 12VDC , RS485 และ Input-Key 1-3 (เหมือนจุดต่อ Input-Key 1-3) เป็นทางเลือกเพื่อความสะดวก โดยต่อผ่านเพียงขั้วเดียว

#### ภาพแสดงบอร์ด และแนวทางการต่อเป็นชุดตัวเลข



## แนวทางการต่อใช้งาน



## การตั้ง Address ด้วย Dip-Switch

บอร์ดหลักจะสามารถตั้ง Address ประจำบอร์ดได้ โดยตั้งผ่าน Dip-Switch 1-4 เพื่อให้เป็น Address 1-15 วิธีการตั้งให้ทำความเข้าใจจากตัวอย่างดังนี้

	Dip-1,+8	Dip-2,+4	Dip-3,+2	Dip-4,+1	
ตัวอย่าง 1	On	Off	Off	On	คือ $8+1 = 9$
ตัวอย่าง 2	Off	On	On	Off	คือ $4+2 = 6$
ตัวอย่าง 3	On	On	Off	On	คือ $8+4+1 = 13$

นอกจากการตั้ง Address ด้วย Dip-Switch แล้ว ยังสามารถตั้งได้จากคำสั่ง :X ผ่านทาง RS485 ด้วย ทั้งนี้จะต้องกำหนดให้ Dip1-4 เป็น Off ทั้งหมด จึงจะทำให้คำสั่ง :X มีผลต่อการใช้งาน

## การตั้ง Digit ด้วย Jumper

บอร์ดหลัก 1 บอร์ด จะสามารถพ่วงตัวเลข Digit ได้ตามต้องการ ซึ่งพ่วงด้วยสายแพผ่านขั้ว 10 Pin โดยจะเพิ่มตัวเลขได้สูงสุด 8 หลัก ทั้งนี้ตัวเลข Digit จะเริ่มจาก 0 ที่หลักขวาสุด และไล่ไปเรื่อย ๆ ทางด้านซ้ายจนถึง 7 ตัว Jumper ทั้ง 3 คือ A2,A1,A0 จะเลือกไปได้ 2 ด้านคือ GND และ +5V วิธีการตั้งให้ทำความเข้าใจจากตัวอย่างดังนี้

	A2,+4	A1,+2	A0,+1	
ตัวอย่าง 1	GND	GND	GND	คือ Digit-0
ตัวอย่าง 2	GND	+5V	GND	คือ $2 = \text{Digit-2}$
ตัวอย่าง 3	+5V	GND	+5V	คือ $4+1 = \text{Digit-5}$
ตัวอย่าง 4	+5V	+5V	+5V	คือ $4+2+1 = \text{Digit-7}$

## การใช้งานทั่วไปและชุดคำสั่ง RS485 แบบ SAC

เมื่อจ่ายไฟเข้าบอร์ด LED สีเหลืองจะกระพริบ 2 ครั้งอย่างรวดเร็ว และพร้อมรับคำสั่งทาง RS485 ทันที 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

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 ส่วนค่าภายใน [ ] หมายถึง Option คือเลือกใส่หรือไม่ใส่ก็ได้ ซึ่งความหมายส่วนใหญ่ก็คือ การใส่ Option คือการ “เขียนค่า” (write) และการไม่ใส่คือการ “อ่านค่า” (read) ในวงเล็บ (R) หมายถึงรับข้อมูลจากตัวแม่ และ (T) หมายถึงข้อมูลที่ส่งกลับไปยังตัวแม่

---

(R) :1xx...x

(T) OK

Set Display (Number,Character)

xx...x คือตัวเลขหรือตัวอักษรที่จะให้แสดงบน Display ซึ่งเป็นได้ทั้งเลข 0-9 และอักษร A-Z ด้วย ทั้งนี้กรณีอักษร A-Z จะเป็นไปเท่าที่เหมาะสมเนื่องจากขีดจำกัดของ 7-Segment เอง นอกจากนี้ยังรับอักขระพิเศษเหล่านี้ด้วย คือ

Blank คือดับหมดทุก Segment

‘-’ ขีดกลาง

‘\_’ ขีดล่าง

‘[’ ปีกกาเปิด

‘]’ ปีกกาปิด

‘.’ จุดทศนิยม

คำสั่ง 1 นี้จะแสดงผลแบบขีดขวา เช่นถ้าสั่งเป็น :14.56 จะปรากฏตัวเลข 4.56 ที่ Digit 2,1,0 จำนวน 3 หลัก และทุกครั้งที่ได้รับคำสั่งนี้ จะมีการ Clear Display ทั้งหมดก่อนเสมอ นั่นหมายความว่า ถ้าเราสั่งเป็น :1 โดยไม่มีข้อมูลเพิ่มเติม ก็จะหมายถึง การ Clear Display ทั้งหมดนั่นเอง

(R) :2nhh...hh

(T) OK

Set Display (Segment)

n คือหมายเลข Digit เริ่มต้น 0-7 , hh...hh คือค่า Hex แต่ละ Byte ที่ต้องการกำหนดให้ Segment ดับหรือติดสว่าง โดยสามารถใส่ต่อเนื่องไปหลาย ๆ Byte ได้ คำสั่งนี้ทำให้ผู้ใช้กำหนดการแสดงผลแต่ละขีด (Segment) ได้อย่างอิสระ การจัดวาง Segment กับเลข Hex ขอให้ทำความเข้าใจในหัวข้อ “รับข้อมูลทาง 3B Port” อีกที

ตัวอย่างเช่น ถ้าส่งคำสั่งเป็น :251C3F5B ก็แสดงผลเป็น u02 ที่ Digit 5,4,3 ข้อมูล Hex จะใส่เป็นที่หลักก็ได้ และจะเรียงจาก Digit มากไปน้อยเสมอ คำสั่ง 2 นี้จะไม่ Clear Display ก่อน นั่นหมายความว่า ถ้าเราส่งคำสั่งไปแสดงผลที่ Digit 5,4,3 แล้ว จากนั้นส่งคำสั่งไปที่ Digit 2,1,0 อีกที สิ่งที่แสดงบน Digit 5,4,3 ก็ยังอยู่เหมือนเดิม

---

(R) :3[xxx]

(T) xxx[OK]

Read/Write I/O ... สำหรับการอ่านหรือเขียนค่าให้กับ I/O ที่ขั้ว 5 Pin โดย xxx คือสถานะของ ขาสัญญาณ P6.1 P6.0 P4.4 ตามลำดับ ซึ่งเป็นสถานะ Logic 0 หรือ 1 โดยตรงเลย

---

(R) :X[aa]

(T) aa[OK]

Read/Write Node Address ... โดย aa คือค่า Address 01-99

---

(R) :Y[aaaa]

(T) ข้อความต่าง ๆ

Checksum or Dump Flash ... สำหรับแสดงค่า Size และ Checksum ของ Firmware เพื่อตรวจสอบความถูกต้อง และถ้าใส่ aaaa คือเลข hex ของ address ก็จะได้แสดงข้อมูลในหน่วยความจำ 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 แต่อยู่ก่อนรหัสส่งท้าย \r (0x0D) จะหมายถึงไม่ต้องการให้มีการตอบกลับใด ๆ โดยที่เครื่องยังคงทำงานตามคำสั่งทุกประการ ซึ่งมักจะใช้กับคำสั่งเพื่อการตั้งค่าหรือแสดงข้อมูล และไม่ต้องการให้ตอบกลับเพื่อความรวดเร็วในการทำงาน ... และในกรณีใช้งานในที่ที่มีสัญญาณรบกวนสูง อาจจะต้องการตรวจสอบคำสั่งให้มีความแน่นอนมากขึ้น ก็สามารถใส่ค่า Checksum เพิ่มเติมได้ตามตัวอย่างนี้

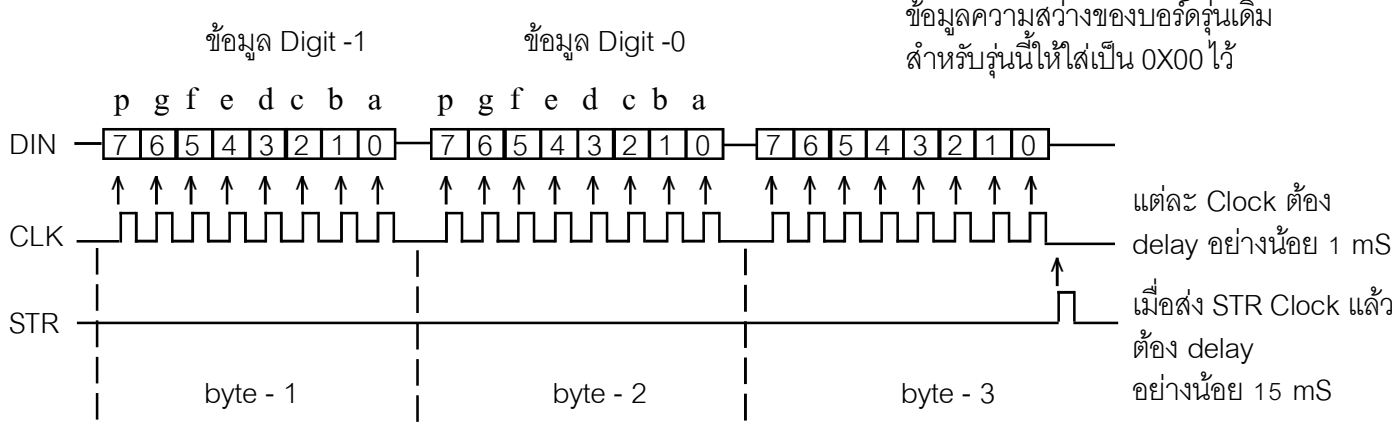
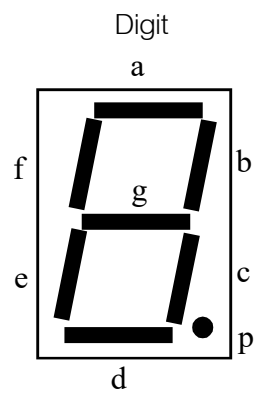
```
:Z@5A\r
EM-D40A v2.2 (AB-603,AB-604)@00\r
```

ค่า Checksum คือผลบวก Hex ของอักษรทุกตัวหลังจาก : จนถึง @ ตัวอย่าง @5A คือค่า Checksum ของอักษร Z และ @00 คือค่า Checksum ของ EM-D40A v2.2 (AB-603,AB-604) ทั้งนี้ถ้าค่า Checksum ไม่ถูกต้อง เครื่องก็จะไม่ทำงานใด ๆ สำหรับข้อมูลที่ตอบกลับก็เช่นกัน ถ้าตรวจสอบค่า Checksum แล้วไม่ถูกต้อง ก็ให้ถือว่าเป็นข้อมูลที่ไม่มีมีความหมายใด ๆ เพราะน่าจะมีผิดพลาดระหว่างการสื่อสารนั่นเอง

**การรับข้อมูลผ่านทาง 3B-Port แบบ 595 Low Speed**

บอร์ดหลักยังสามารถรับข้อมูลทาง 3B Port ได้ด้วย การใช้งานในโหมดนี้ จะต้องใส่ต่อสาย Input-Key4 กับ Gnd ไว้ (หรืออาจจะเอาชิพ PC817 ออก แล้วใช้ลวด Jump ขา 3,4 ก็ได้) ทั้งนี้เมื่อจ่ายไฟเลี้ยง บอร์ดก็จะเข้าสู่โหมด 3B-Port ทันที รูปแบบสัญญาณสื่อสาร จะเป็นทำนองเดียวกับชิพ TPIC6B595 แต่เป็น Low Speed คือต้องมีการ Delay มากเพียงพอ (ดูรายละเอียดจากภาพ) ขาสัญญาณทาง 3B-Port คือดังนี้ P6.1 = CLK , P6.0 = STR และ P4.4 = DIN ลักษณะของข้อมูล จะแยกเป็นแต่ละ Byte (8 Bit) ผ่านขาสัญญาณ DIN (0=มืด 1=สว่าง) และ CLK (กระทำที่ขอบขาขึ้น) และเมื่อส่งจนครบทุก Byte ตามต้องการแล้ว ก็จึงส่งสัญญาณ STR (กระทำที่ขอบขาขึ้น) เพื่อให้ทำการแสดงผล ข้อมูลแต่ละ Byte ที่ส่งก็คือข้อมูล Segment ของแต่ละหลักโดยจะเรียงจาก Digit มากไปน้อย (ขวาไปซ้าย) ทั้งนี้ Byte ก่อนสุดท้ายก็คือ Digit-0 นั่นเอง ส่วน Byte สุดท้ายสำหรับสินค้านี้เดิม มันรหัสควบคุมความสว่าง แต่สำหรับรุ่นใหม่นี้ไม่จำเป็นต้องกำหนดความสว่างใด ๆ จึงให้ส่งเป็น 0x00 ไว้แทน รูปแบบสัญญาณขอให้ทำความเข้าใจจากภาพต่อไปนี้

**ภาพสัญญาณ 3B Port**

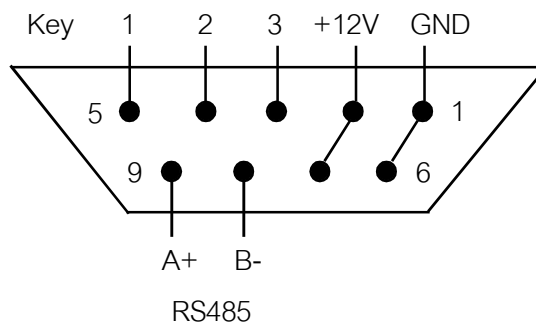


การรับข้อมูลทาง 3B-Port นี้ จะรองรับจำนวน Byte ได้สูงสุด 16 หลัก (แตกต่างจากทาง RS485 ที่เป็น 8 หลัก) นั้นหมายถึงจะใช้ชุดตัวเลขแบบ 8 หลัก 2 ชุดโดยรับข้อมูลทาง 3B-Port เดียวกันได้ นั่นจึงเป็นเหตุให้หัว 5 Pin มีไว้ 2 หัว เพื่อให้ต่อพ่วงได้สะดวกนั่นเอง และใช้ Dip-Switch กำหนดหมายเลขของหลักขวาสุด ที่จะแสดงผลนั่นเอง ขอทำความเข้าใจจากตัวอย่างดังนี้

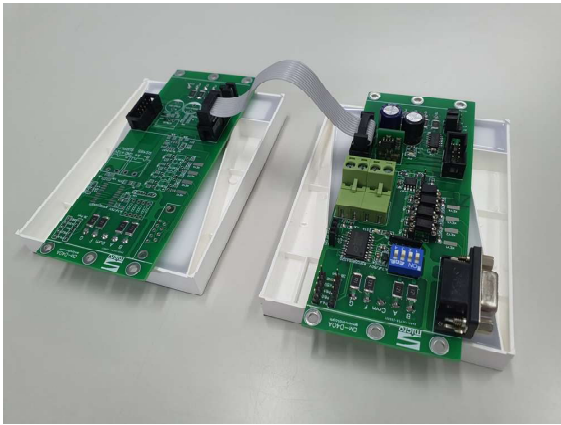
ตัวอย่างการส่งข้อมูล 1 ส่ง 4 byte คือ 0x1c 0x3f 0x5b 0x00  
 ตั้ง Dip-Switch ไว้ที่ 0 ... จะแสดงเป็น u02  
 ตั้ง Dip-Switch ไว้ที่ 2 ... จะแสดงเป็น u

ตัวอย่างการส่งข้อมูล 2 ส่ง 7 byte คือ 0x06 0x5b 0x4f 0x66 0x6D 0x7D 0x00  
 ตั้ง Dip-Switch ไว้ที่ 0 ... จะแสดงเป็น 123456  
 ตั้ง Dip-Switch ไว้ที่ 2 ... จะแสดงเป็น 1234  
 ตั้ง Dip-Switch ไว้ที่ 4 ... จะแสดงเป็น 12

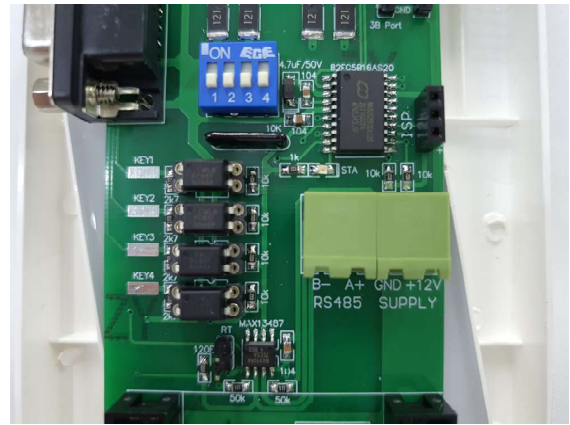
หัว DB9 ตัวเมีย สำหรับการต่อใช้งานอย่างสะดวก



# ภาพแสดงส่วนต่างๆ บนบอร์ดหลัก



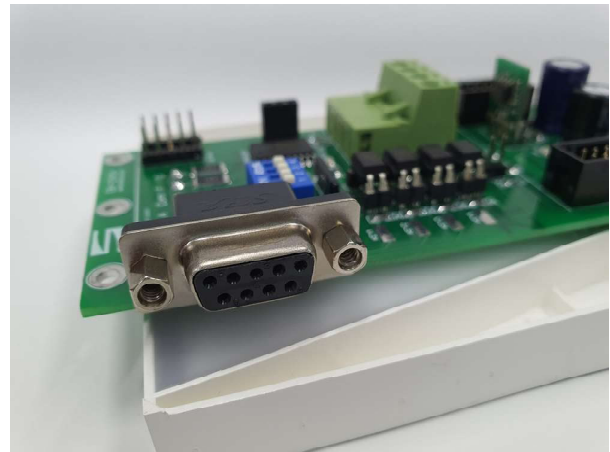
การพ่วง Digit ด้วยสาย 10 Pin ด้านหลัง



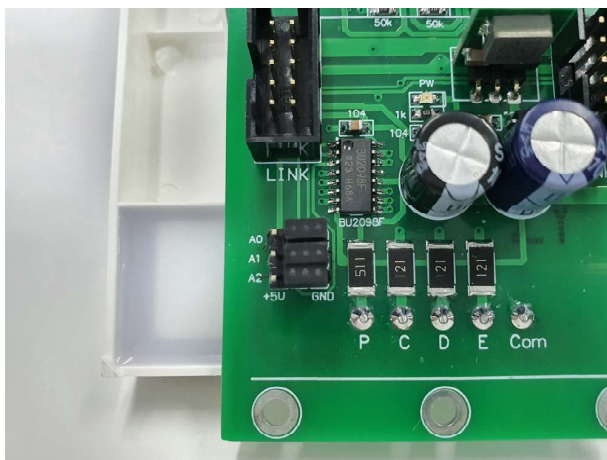
บริเวณหัวต่อไขน็อต  
จุดต่อสำหรับปุ่มกด และ Dip-Switch



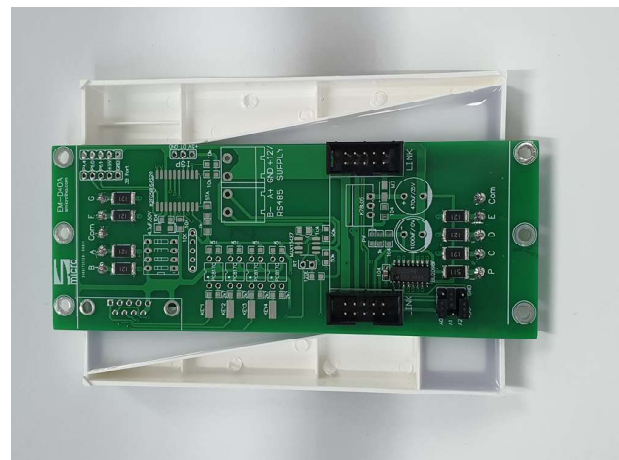
หัว 3B-Port



DB9S เพื่อการต่อใช้งานอย่างสะดวก



Jumper ตั้งเลข Digit  
(ในภาพตั้งเป็นหลักที่ 0)



ภาพด้านหลังของบอร์ด EM-40AD