

4 Inch 7-Segment Module

ส่วนประกอบสินค้า เฉพาะตัวโมดูล ไม่มีภาคจ่ายไฟ

EM-D40 คือโมดูลตัวแสดงผลแบบ 7-Segment LED สีแดง รับคำสั่งควบคุมได้ทั้งจาก RS485 และ 3B Port ใช้ต่อกับเครื่องคอมพิวเตอร์เพื่อการแสดงผลได้อย่างสะดวก หรือจะต่อกับบอร์ดไมโครคอนโทรลเลอร์ก็ได้เช่นกัน ตัวโมดูลออกแบบให้ต่อพ่วง Digit กันได้อย่างสะดวก และแยกย่อยเป็นส่วน ๆ เพื่อเลือกใช้งานดังนี้

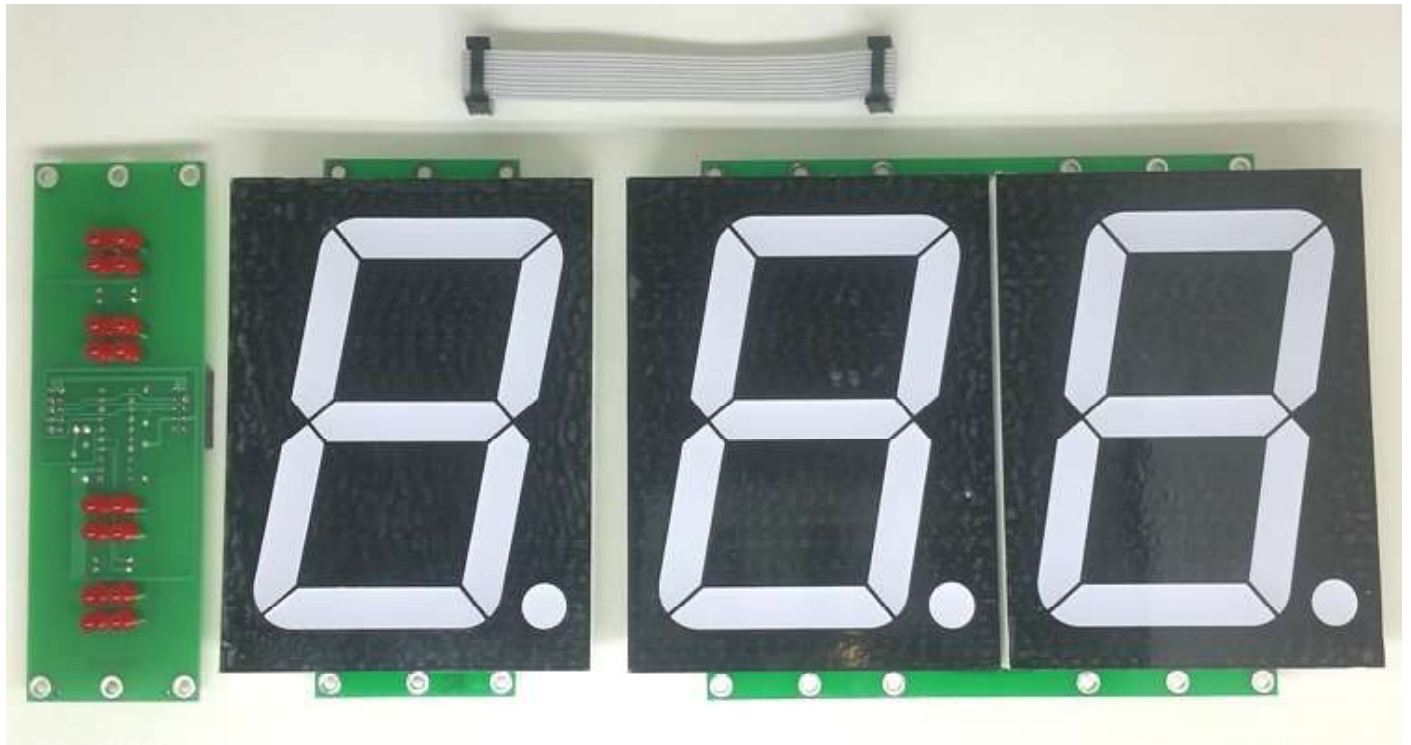
- EM-D40M บอร์ดหลัก (Main) ที่มี MCU ในตัวและมีตัวเลข 2 หลัก
- EM-D40D บอร์ดตัวเลขแต่ละหลัก ที่ต่อพ่วงขยายไปได้ตามต้องการ
- EM-D40C บอร์ดแสดงเครื่องหมาย Colon จุดหรือองศา โดยใช้ LED มาจัดวาง

เลือกต่อพ่วงได้สูงสุดถึง 16 หลัก (รวม 2 ตัวเลขในบอร์ดหลักด้วย) โดยจะจัดวางเฉพาะตัวเลข (Digit) หรือจะแทรก Colon ไว้ตรงไหนก็ได้ตามต้องการ สัญญาณควบคุมและไฟเลี้ยงจะพ่วงไปพร้อมกันด้วยชุดสายแพ 10 เส้นด้านหลัง สามารถปรับความสว่างของ LED ได้ 5 ระดับ (ใช้ภายในอาคาร In-Door) ช่วยประหยัดพลังงาน กรณีสั่งงานผ่าน RS485 ยังสามารถต่อพ่วงทางด้าน RS485 ได้ โดยตั้ง Node Address ได้ 2 หลัก ทำให้ควบคุมการแสดงผลได้เป็นจำนวนมากผ่านพอร์ท RS485 เดียว ประยุกต์ใช้งานได้หลากหลายสุดยอด

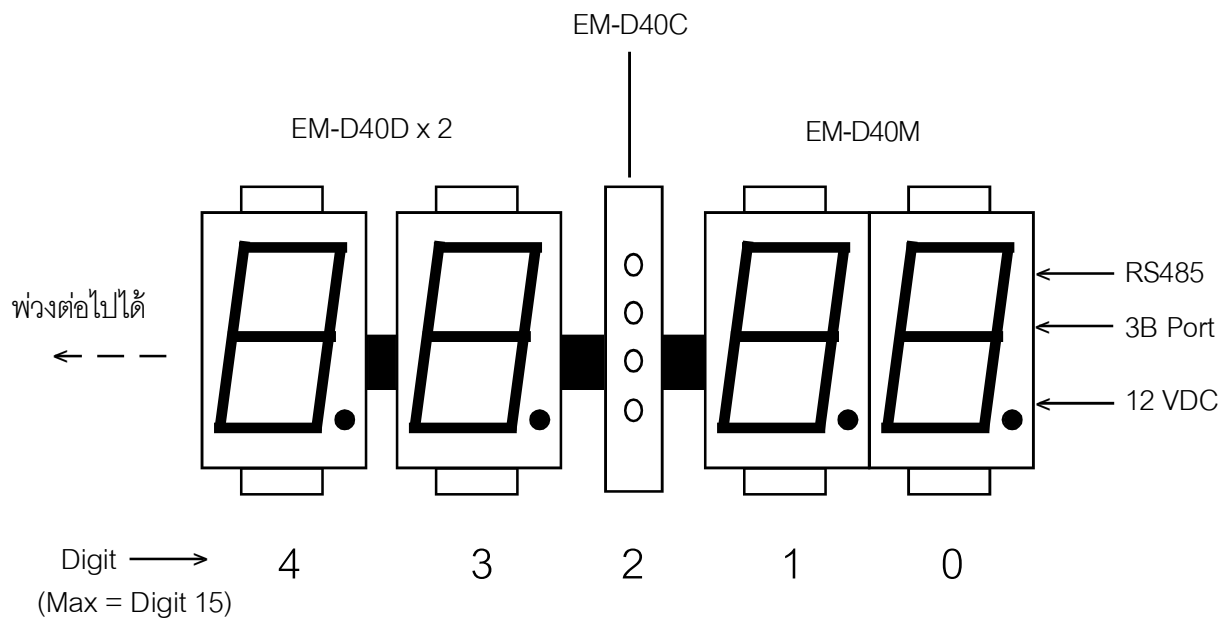
คุณสมบัติ

- ทำงานด้วยไมโครคอนโทรลเลอร์เบอร์ 87FE6051 ความถี่ภายใน 11.0592 MHz
- ตัวเลข 7-Segment LED ความสูง 4 นิ้ว ปรับความสว่างได้ 5 ระดับ (0-4) ใช้ภายในอาคาร (In-Door)
- ต่อพ่วงตัวเลข (Digit) หรือ Colon ได้สูงสุด 16 หลัก (รวม 2 ตัวเลขในบอร์ดหลักด้วย)
- การต่อพ่วงทำด้วยชุดสายแพ 10 เส้นด้านหลัง ซึ่งจะมีสัญญาณควบคุมและไฟเลี้ยงไปพร้อมกัน
- มี RS485 เพื่อการรับคำสั่งควบคุม และมี Jumper เลือกใช้ R-Terminate สำหรับปลายสายได้
- คุณสมบัติการสื่อสารคือ Speed=9600 Parity=None Data=8 Stop=1
- สามารถตั้ง Node Address ได้ 2 หลัก ทำให้ใช้งานต่อพ่วงกันเป็น RS485 Network ได้อีกด้วย
- ขั้ว 5 Pin สำหรับรับข้อมูลแบบ 3 Bit คือ Data-In Clock และ Strobe โดยมีลักษณะสัญญาณเหมือนกับชิพยอคนิยม TPIC6B595 แต่จะเป็นแบบ Low Speed (ส่งข้อมูล 16 หลักใช้เวลา 0.14 วินาที)
- ใช้ไฟเลี้ยง 12 VDC และมี LED สีแดงแสดงสถานะไฟเลี้ยงในบอร์ดหลัก
- ขั้วต่อ RS485 และไฟเลี้ยงแบบไขว้กัน เสียบและดึงออกจากบอร์ดได้สะดวกมาก
- บอร์ด EM-D40M ขนาดกว้าง 182 สูง 137 และหนา 40 mm และกินกระแสสูงสุด 560 mA
- บอร์ด EM-D40D ขนาดกว้าง 91 สูง 137 และหนา 40 mm และกินกระแสสูงสุด 215 mA
- บอร์ด EM-D40C ขนาดกว้าง 45 สูง 137 และหนา 40 mm และกินกระแสสูงสุด 65 mA

ภาพแสดงบอร์ด



ภาพการต่อพ่วง



การใช้งานทั่วไป และคำสั่งควบคุมแบบ Sac

เมื่อจ่ายไฟเข้าบอร์ด EM-D40 จะพร้อมใช้งานภายใน 1/2 วินาที โดยพร้อมจะรับข้อมูลทั้งทาง RS485 และทาง 3B Port ด้วย Node Address ของบอร์ดจะมีค่า Default เป็น 00 (คือไม่ใช้งาน) และค่าความสว่างเป็น 4 (คือสว่างสุด) ... สำหรับการสื่อสารผ่าน RS485 เพื่อส่งคำสั่งมาควบคุมนั้น ให้ตั้งคุณสมบัติเป็น Speed=9600 Parity=None Data=8 Stop=1 ... ชุดคำสั่งที่ใช้จะเป็นแบบ Sac Protocol คือ Smart Ascii Command สามารถใช้โปรแกรมสื่อสารแบบ Terminal ที่ทำงานบนเครื่องคอมพิวเตอร์ เพื่อการทดสอบคำสั่งได้ทันที เพราะมีรูปแบบเป็นตัวอักษร Ascii ทั้งหมด ชุดคำสั่งจะมาจากส่วนกลาง (Master) ซึ่งอาจเป็นคอมพิวเตอร์หรือบอร์ดไมโครคอนโทรลเลอร์ก็ได้ โดยมีรูปแบบดังนี้...

:c\r สำหรับใช้งานแบบไม่มี Node Address คือใช้งานเพียงบอร์ดเดียว
:@aac\r สำหรับใช้งานแบบมี Node Address คือใช้พ่วงกันหลาย ๆ บอร์ด

: คือรหัสหน้าหน้า (0x3A)

c คือรหัสคำสั่งเป็นตัวเลข หรือตัวอักษรตัวเดียว สำหรับ EM-D40 คือ 1,2,3,X,Y,Z

\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 ไม่จำเป็นต้องตั้งผ่าน Dip-Switch หรือด้วยปุ่มกดใด ๆ 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

ชุดคำสั่งของ EM-D40 จะสรุปได้ตามตารางต่อไปนี้ โดยจะแสดงในรูปแบบไม่มี Node Address และไม่ใส่รหัสลงท้าย \r เพื่อให้ดูสบายตา ส่วนค่าภายใน [] หมายถึง Option คือเลือกใส่หรือไม่ใส่ก็ได้ ซึ่งความหมายส่วนใหญ่ก็คือ การใส่ Option คือการ “เขียนค่า” (write) และการไม่ใส่คือการ “อ่านค่า” (read) ถ้าใน Option มีทางเลือกหลายแบบ ก็จะขึ้นด้วยอักษร | อีกทีในวงเล็บ (RX) หมายถึง EM-D40 เป็นฝ่ายรับข้อมูล และ (TX) หมายถึง EM-D40 เป็นฝ่ายส่งข้อมูล

รูปแบบคำสั่งและการใช้งาน

รูปแบบคำสั่ง	การใช้งาน
(RX) :1xx...x (TX) OK	<p>Set Display (Number,Character) xx...x คือตัวเลขหรือตัวอักษรที่จะให้แสดงบน Display ซึ่งเป็นได้ทั้งเลข 0-9 และอักษร A-Z ด้วย ทั้งนี้กรณีอักษร A-Z จะเป็นที่เหมาะที่สุด เนื่องจากขีดจำกัดของ 7-Segment เอง นอกจากนี้ยังรับอักขระพิเศษเหล่านี้ด้วย คือ</p> <p>Blank คือดับหมดทุก Segment `·` ขีดกลาง `·` ขีดกลาง `(` ปีกกาเปิด `)` ปีกกาปิด `·` จุดทศนิยม (บนตัวเลข) `·` แสดง Colon บน EM-D40C (ที่ต้องใช้ Semi-Colon แทน เนื่องจากเครื่องหมาย : เป็นรหัสนำหน้าคำสั่ง จึงต้องใช้เป็น ; แทน)</p> <p>คำสั่ง 1 นี้จะแสดงผลแบบขีดขวา เช่นถ้าสั่งเป็น :14.56 ก็จะปรากฏตัวเลข 4.56 ที่ Digit 2,1,0 จำนวน 3 หลัก และทุกครั้งที่ได้รับคำสั่งนี้ จะมีการ Clear Display ทั้งหมดก่อนเสมอ นั่นหมายถึงว่า ถ้าเราสั่งเป็น :1 โดยไม่มีข้อมูลเพิ่มเติม ก็จะหมายถึงการ Clear Display ทั้งหมดนั่นเอง</p>
(RX) :2nhh...hh (TX) OK	<p>Set Display (Segment) n คือหมายเลข Digit เริ่มต้น 0-F (เลข Hex) hh...hh คือค่า Hex แต่ละ Byte ที่ต้องการกำหนดให้ Segment ดับหรือติดสว่าง โดยสามารถใส่ต่อเนื่องไปหลาย ๆ Byte ได้ คำสั่งนี้ทำให้ผู้ใช้กำหนดการแสดงผลแต่ละขีด (Segment) ได้อย่างอิสระ การจัดวาง Segment กับเลข Hex ขอให้ทำความเข้าใจจากภาพในหัวข้อ "ส่งข้อมูลทาง 3B Port" อีกที</p> <p>ตัวอย่างเช่น ถ้าส่งคำสั่งเป็น :291C3F5B ก็จะแสดงผลเป็น u02 ที่ Digit 9,8,7 ข้อมูล Hex จะใส่เป็นที่หลักก็ได้ และจะเรียงจาก Digit มากไปน้อยเสมอ คำสั่ง 2 นี้จะไม่ Clear Display ก่อน นั่นหมายความว่า ถ้าเราส่งคำสั่งไปแสดงผลที่ Digit 9,8,7 แล้วยังคงค้างคำสั่งไปที่ Digit 2,1,0 อีกที สิ่งที่จะแสดงบน Digit 9,8,7 ก็ยังคงอยู่เหมือนเดิม</p>
(RX) :3(b) (TX) b(OK)	<p>Set Bright b คือค่าความสว่างของ Display เป็น 0-4 โดย 0 คือสว่างน้อยสุด และ 4 คือสว่างมากที่สุด (Default เป็น 4) เมื่อตั้งค่าจากคำสั่งนี้แล้ว ค่าความสว่างจะจำไว้ในบอर्डเลย เมื่อปิดเปิดใหม่ ก็ยังคงเป็นค่าล่าสุดที่ได้ตั้งไว้</p>
(RX) :X(aa) (TX) aa(OK)	<p>Read/Write Node Address aa คือค่า Address 01-99 (bcd)</p>
(RX) :Y(aaaa) (TX) ขอความต่าง ๆ	<p>Checksum or Dump Flash สำหรับแสดงค่า Size และ Checksum ของ Firmware เพื่อตรวจสอบความถูกต้อง และถ้าใส่ aaaa คือเลข hex ของ address ภายนอก จะเป็นการแสดงค่าข้อมูลใหญ่ด้วย คือข้อมูลใน Flash ของตัว MCU นั้นเอง โดยจะแสดงเป็นจำนวน 128 Byte</p>
(RX) :Z(!) (TX) ขอความต่าง ๆ	<p>Show Model / Version & Self-Test ถ้าใส่ ! จะหมายถึงเขาระบบ Self-Test</p>

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

คำสั่งควบคุมแบบ Cnet

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

:Z@5Ar

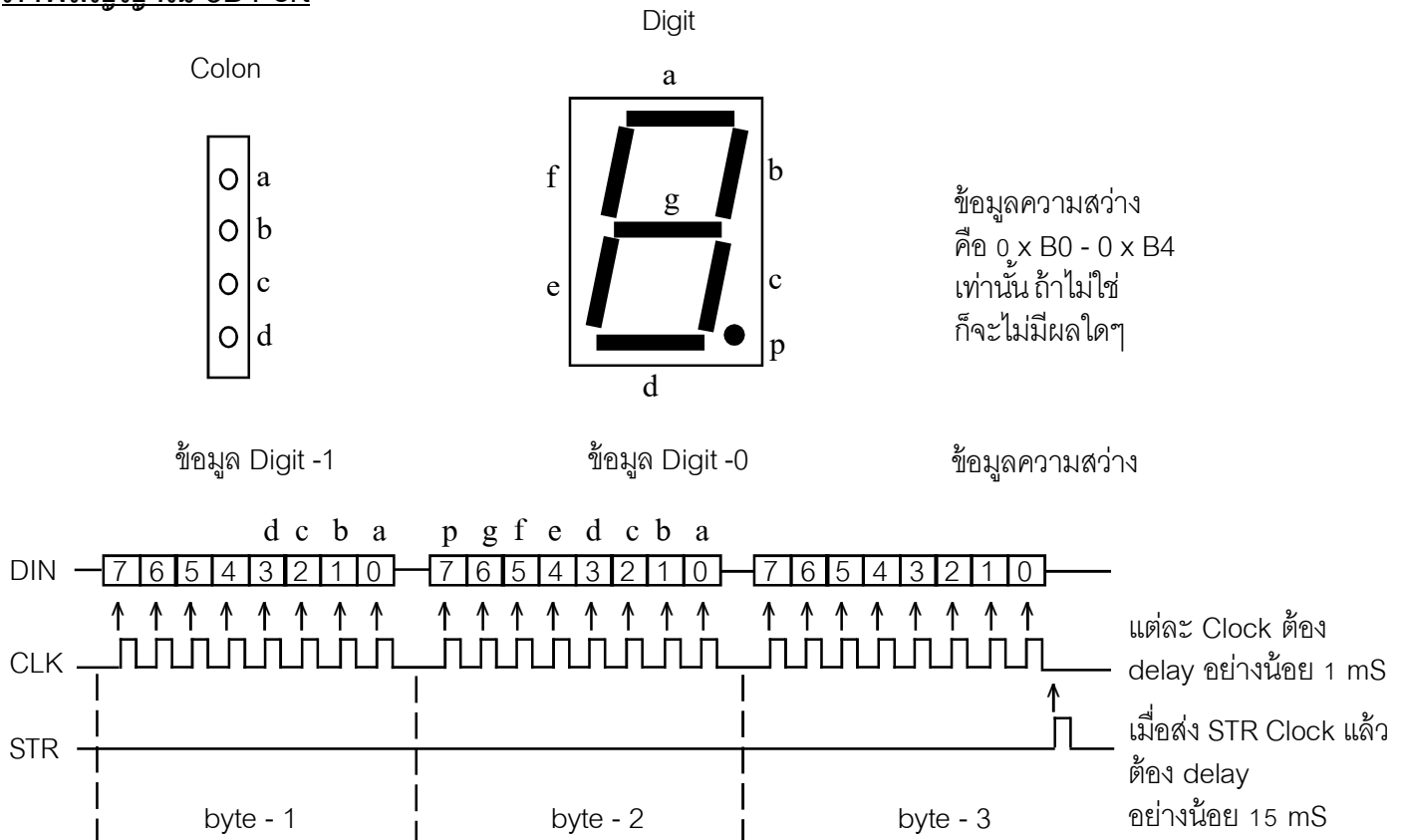
EM-D23,40 v1.2 (2017)@5Ar

ค่า Checksum คือผลบวก Hex ของอักษรทุกตัวหลังจาก : จนจบคำสั่ง จากตัวอย่าง @5A คือค่า Checksum ของอักษร Z และ @5A คือค่า Checksum ของ EM-D23,40 v1.2 (2017) ทั้งนี้ถ้าค่า Checksum ไม่ถูกต้อง เครื่องก็จะไม่ทำงานใด ๆ สำหรับข้อมูลที่ตอบกลับก็เช่นกัน ถ้าทางด้านคอมพิวเตอร์ตรวจสอบค่า Checksum แล้วไม่ถูกต้อง ก็ให้ถือว่าเป็นข้อมูลที่ไม่มี ความหมายใด ๆ เพราะน่าจะมีความผิดพลาดระหว่างการสื่อสารนั่นเอง

ส่งข้อมูลทาง 3B Port

EM-D40 สามารถรับข้อมูลทาง 3B Port ได้ด้วย โดยรูปแบบสัญญาณเป็นไปในทำนองเดียวกับชิพ TPIC6B595 แต่จะต้องมีการ Delay มากเพียงพอ (ดูรายละเอียดจากภาพ) ลักษณะของข้อมูลจะแยกเป็นแต่ละ Byte (8 Bit) ผ่านขาสัญญาณ DIN (0=มืด 1=สว่าง) และ CLK (กระทำที่ขอบขาขึ้น) และเมื่อส่งจนครบทุก Byte ตามต้องการแล้ว ก็จึงส่งสัญญาณ STR (กระทำที่ขอบขาขึ้น) เพื่อให้ทำการแสดงผล ข้อมูลแต่ละ Byte ที่ส่งก็คือข้อมูล Segment ของแต่ละหลัก โดยจะเรียงจาก Digit มากไปน้อย (ขวาไปซ้าย) ทั้งนี้ Byte ก่อนสุดท้ายก็คือ Digit-0 นั่นเอง ส่วน Byte สุดท้ายจะเป็นรหัสควบคุมความสว่าง ซึ่งจะต้องมีค่าเป็น 0xB0 ถึง 0xB4 เท่านั้น จึงจะเป็นการตั้งความสว่าง ถ้าส่งเป็นค่าอื่น ๆ ก็จะไม่แสดงผลอะไร การตั้งความสว่างจาก 3B Port จะมีผลในขณะนั้น แต่จะไม่มีผลซ้ำค่าแต่อย่างใด ซึ่งแตกต่างจากคำสั่ง :3 ที่สั่งจาก Sac Protocol ซึ่งจะจำค่าความสว่างไว้ด้วยรูปแบบสัญญาณขอให้ทำความเข้าใจจากภาพต่อไปนี้

ภาพสัญญาณ 3B Port



ตัวอย่างการส่งข้อมูล

(1) ส่ง 4 byte คือ 0x1c 0x3f 0x5b 0x00
จะแสดงเป็น u02 และไม่มีการตั้งความสว่าง

(2) ส่ง 3 byte คือ 0x66 0x6d 0xb4
จะแสดงเป็น 45 และตั้งความสว่างเป็นระดับ 4 (สว่างสุด)

(3) ส่ง 1 byte คือ 0xb2
จะไม่มีการเปลี่ยนแปลงใดๆ ยกเว้นปรับความสว่างเป็นระดับ 2

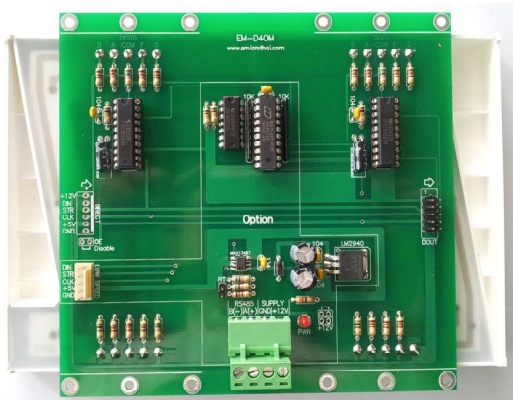
ภาพแสดงส่วนต่างๆ



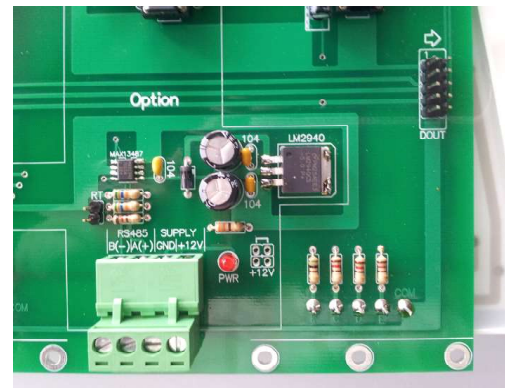
แสดงการต่อใช้งาน



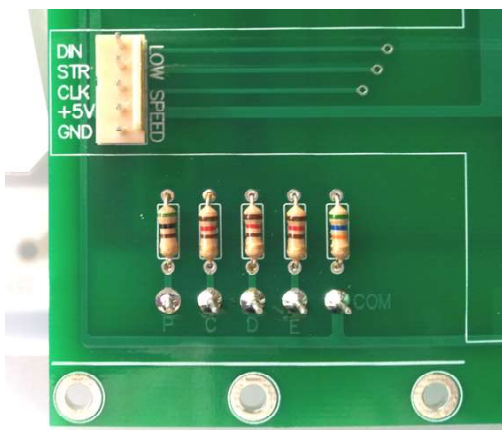
บอร์ดหลัก



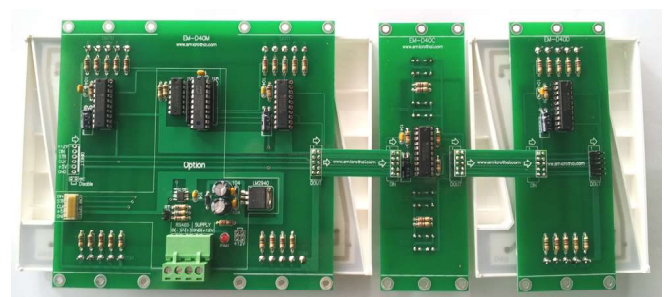
ด้านหลังของบอร์ดหลัก



บริเวณขั้วต่อและขาต่อพ่วง



ขั้วต่อ 3B Port



การต่อพ่วงมองจากด้านหลัง

