I ไดรเวอร์การเชื่อมต่อผ่านหน่วยความจำ

1 วิธีการเชื่อมต่อผ่านหน่วยความจำ 3
2 คำสั่งของการเชื่อมต่อผ่านหน่วยความจำ 17
3 รายละเอียดของระบบ 46
4 การเลือกไดรเวอร์ 47
5 ตัวอย่างการตั้งค่าการสื่อสาร 48
6 รายการตั้งค่า 56
7 แผนภาพแสดงการต่อสายเคเบิล 62
8 อุปกรณ์ที่รองรับ 78
9 รหัสอุปกรณ์และรหัสตำแหน่ง 79
10 ข้อความแสดงข้อผิดพลาด 80
11 API การเชื่อมต่อผ่านหน่วยความจำ (การสื่อสารแบบ Ethernet) 83
12 โปรแกรมตัวอย่าง (การสื่อสารแบบอนุกรม) 115
13 โปรแกรมตัวอย่าง (การสื่อสารแบบ Ethernet) 129

ข้อมูลเบื้องต้น

คู่มือนี้จะอธิบายถึงวิธีเชื่อมต่อจอแสดงผล (GP3000 series) เข้ากับอุปกรณ์ภายนอก (PLC เป้าหมาย) โดยคุณสามารถดูคำอธิบายขั้นตอนการเชื่อมต่อได้ในส่วนต่าง ๆ ต่อไปนี้

1	**รายละเอียดของระบบ** **ส่วนนี้จะแสดงชนิดของอุปกรณ์ภายนอกที่สามารถเชื่อมต่อได้และชนิดของ SIO**	☞ **"รายละเอียดของระบบ"** (หน้า 46)
2	**การเลือกอุปกรณ์ภายนอก** **เลือกรหัส (รุ่น) ของอุปกรณ์ภายนอกที่จะเชื่อมต่อและวิธีเชื่อมต่อ**	☞ **"การเลือกไดรเวอร์"** (หน้า 47)
3	**ตัวอย่างการตั้งค่าการสื่อสาร** **ส่วนนี้จะแสดงตัวอย่างการตั้งค่าสำหรับการสื่อสารระหว่างจอแสดงผลและอุปกรณ์ภายนอก**	☞ **"ตัวอย่างการตั้งค่าการสื่อสาร"** (หน้า 48)
4	**รายการตั้งค่า** **ส่วนนี้จะอธิบายถึงรายการตั้งค่าการสื่อสารบนจอแสดงผล** **ทั้งการตั้งค่าด้วย GP-Pro Ex หรือการตั้งค่าในโหมดออฟไลน์**	☞ **"รายการตั้งค่า"** (หน้า 56)
5	**แผนภาพแสดงการต่อสายเคเบิล** **ส่วนนี้จะแสดงสายเคเบิลและอะแดปเตอร์ต่าง ๆ ที่ใช้เชื่อมต่อจอแสดงผลและอุปกรณ์ภายนอก**	☞ **"แผนภาพแสดงการต่อสายเคเบิล"** (หน้า 62)

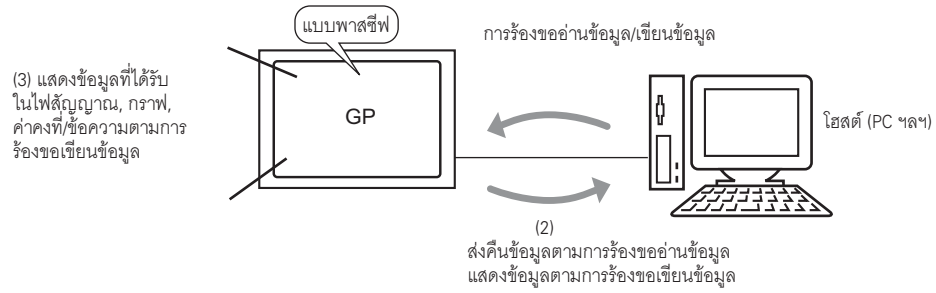
การดำเนินการ

1 วิธีการเชื่อมต่อผ่านหน่วยความจำ

ใน “วิธีการเชื่อมต่อผ่านหน่วยความจำ” การร้องขออ่านข้อมูล/เขียนข้อมูลจะเกิดขึ้นจากเครื่องโฮสต์ไปยัง GP ตามภาพดังต่อไปนี้ GP จะแสดงข้อมูลบนหน้าจอที่ถูกส่งไปตามการร้องขอเขียนข้อมูลของโฮสต์ ในการตอบสนองต่อการร้องขออ่านข้อมูล เครื่อง GP จะส่งข้อมูลที่จัดเก็บไว้ภายใน GP ไปยังเครื่องโฮสต์ด้วย

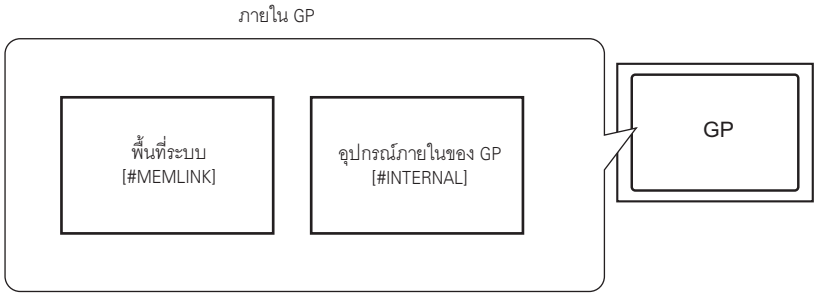
หมายเหตุ

- การสื่อสารที่อิงตามวิธีการเชื่อมต่อผ่านหน่วยความจำจะสำเร็จได้โดยการเรียกใช้โปรแกรมบนโฮสต์

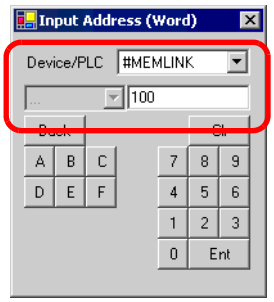


◆ ตำแหน่งที่ใช้งานได้

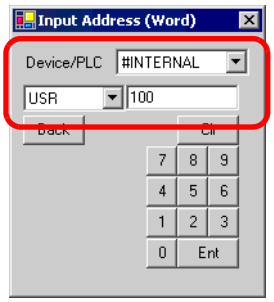
เพื่อให้ GP สามารถรับข้อมูลการแสดงผลที่จำเป็นจากโฮสต์ได้ในขณะสื่อสาร ให้กำหนดตำแหน่งที่สามารถอ้างอิงข้อมูลและตั้งค่าพาร์ทและคุณสมบัติสคริปต์ ตำแหน่งภายใน GP ที่สามารถตั้งค่าเป็นปลายทางข้อมูลอ้างอิงได้มีอยู่ด้วยกัน 2 ชนิด คือ



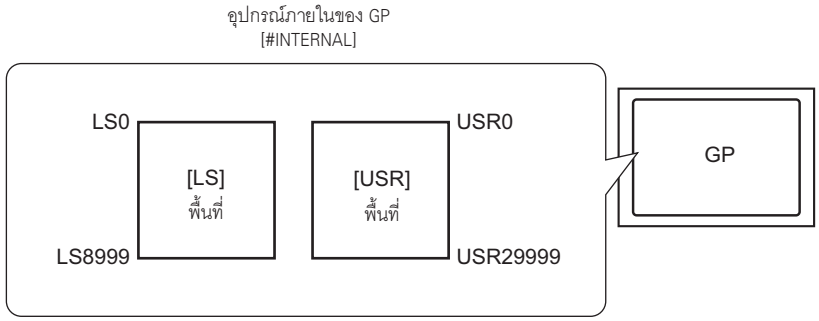
- ตำแหน่งพื้นที่ระบบของการเชื่อมต่อผ่านหน่วยความจำ
  พื้นที่ระบบเป็นสื่อกลางที่ใช้เพื่อร้องขอการอ่าน/การเขียนข้อมูลของโฮสต์ และยังเป็นพื้นที่การสื่อสารของวิธีการเชื่อมต่อผ่านหน่วยความจำอีกด้วย
  สำหรับรายละเอียดเกี่ยวกับพื้นที่ระบบ โปรดดูที่ “พื้นที่ระบบ (พื้นที่การเชื่อมต่อผ่านหน่วยความจำ)” (หน้า 6)
  ยกตัวอย่างเช่น หากต้องการตั้งค่าตำแหน่งของ “สวิตช์ตั้งค่าตำแหน่งเวิร์ด” ให้เลือก [#MEMLINK] จาก [Device/PLC] และป้อนตำแหน่งของรุ่นผลิตภัณฑ์นั้น (เช่น “0100”)
  (ตัวอย่าง หน้าจอ Input Address บน “สวิตช์ตั้งค่าตำแหน่งเวิร์ด”)



- ตำแหน่งอุปกรณ์ภายในของ GP
  ให้กำหนดตำแหน่งชนิดนี้เมื่ออ้างอิงถึงปลายทางของค่าที่คำนวณแล้วซึ่งจัดเก็บไว้ชั่วคราวภายใน GP โดยเลือก [#INTERNAL] (โปรดดูจากอุปกรณ์ภายในของ GP) เป็น [Device/PLC] ซึ่งจะสื่อสารกับ GP และป้อนตำแหน่งนั้น (เช่น “USR00100”)
  (ตัวอย่าง หน้าจอ Input Address บน “สวิตช์ตั้งค่าตำแหน่งเวิร์ด”)



อุปกรณ์ภายใน [#INTERNAL] ของ GP จะมีพื้นที่ซึ่งสร้างขึ้นสองแห่ง ได้แก่ พื้นที่ [LS] และพื้นที่ [USR] (ที่แสดงทางด้านล่าง) อย่างไรก็ตาม วิธีการเชื่อมต่อผ่านหน่วยความจำเพื่อการสื่อสารสามารถใช้ได้เฉพาะพื้นที่ [USR] เท่านั้น



พื้นที่ [LS]

พื้นที่นี้เป็นพื้นที่ที่ผู้ใช้สามารถใช้ได้อย่างอิสระตามต้องการ และเป็นพื้นที่ซึ่งใช้สำหรับการใช้งาน GP

Cf. คู่มืออ้างอิงสำหรับ GP-Pro EX **"**ภาคผนวก 1.4 พื้นที่ LS (เฉพาะวิธีการเชื่อมต่อโดยตรงเท่านั้น)**"**

ข้อสำคัญ
- พื้นที่นี้ใช้สำหรับสื่อสารด้วยวิธีการเชื่อมต่อผ่านหน่วยความจำไม่ได้

พื้นที่ [USR]

พื้นที่นี้มีจำนวนเวิร์ดทั้งสิ้น 30,000 เวิร์ด และสามารถใช้เป็นพื้นที่สำหรับผู้ใช้ได้อย่างอิสระตามต้องการ

1.1 พื้นที่ระบบ (พื้นที่การเชื่อมต่อผ่านหน่วยความจำ)

เมื่อสื่อสารภายใน GP ด้วยวิธีการเชื่อมต่อผ่านหน่วยความจำ เครื่องจะรักษาความปลอดภัยของพื้นที่ LS พื้นที่นี้ถูกใช้สำหรับแลกเปลี่ยนข้อมูลกับโฮสต์

1.1.1 รายการพื้นที่ระบบ

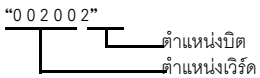
พื้นที่ระบบของวิธีการเชื่อมต่อผ่านหน่วยความจำ

ข้อสำคัญ
- โปรดอย่าตั้งค่าตำแหน่งสำหรับพาร์ทที่ขยายพื้นที่เก็บข้อมูลระบบและพื้นที่สำหรับอ่านข้อมูล หรือพื้นที่สำหรับอ่านข้อมูลและพื้นที่สำหรับผู้ใช้
- เมื่อตั้งค่าตำแหน่งสำหรับพาร์ทในพื้นที่เก็บข้อมูลระบบ โปรดกำหนดความยาวข้อมูลเป็น 16 บิต

ชื่อพื้นที่	คำอธิบาย
พื้นที่เก็บข้อมูลระบบ	พื้นที่นี้จัดเก็บข้อมูลที่จำเป็นสำหรับการทำงานของระบบ เช่น ข้อมูลควบคุมหน้าจอ GP และข้อมูลข้อผิดพลาด รายละเอียดการเขียนจะถูกกำหนดให้ ☞ **"พื้นที่เก็บข้อมูลระบบ"** (หน้า 7)
พื้นที่สำหรับผู้ใช้	พื้นที่นี้ใช้สำหรับแลกเปลี่ยนข้อมูลระหว่าง GP กับโฮสต์ (คอมพิวเตอร์ ฯลฯ) ให้กำหนดที่โฮสต์ว่าจะเขียนข้อมูลตำแหน่ง GP ใด และสร้างโปรแกรมเพื่อเขียนข้อมูล ใน GP ให้กำหนดการตั้งค่าสำหรับพาร์ทพิเศษเพื่อแสดงข้อมูลที่เขียนในตำแหน่งเหล่านั้น เพื่อให้โฮสต์อ่านค่าที่เขียนด้วยสวิตช์พาร์ทแสดงผลข้อมูล และแป้นคีย์ได้ คุณต้องสร้างโปรแกรมในโฮสต์สำหรับอ่านข้อมูลของ GP
รีเลย์พิเศษ	พื้นที่นี้จัดเก็บข้อมูลสถานะแต่ละชนิดที่เกิดขึ้นเมื่อ GP ทำการสื่อสาร ☞ **"รีเลย์พิเศษ"** (หน้า 14)
สำรอง	ใช้ภายใน GP โปรดอย่าใช้พื้นที่นี้ มิฉะนั้น พื้นที่นี้จะทำงานผิดปกติ

หมายเหตุ
- เมื่อตำแหน่งมีการกำหนดบิต ให้เพิ่มตำแหน่งบิตต่อจากอุปกรณ์ชนิดเวิร์ด (กำหนดตั้งแต่ 00 ถึง 15)

ตัวอย่าง เมื่อกำหนดบิต 02 ของตำแหน่ง 0020 ของพื้นที่สำหรับผู้ใช้



1.1.2 พื้นที่เก็บข้อมูลระบบ

แสดงรายละเอียดของข้อมูลที่เขียนในแต่ละตำแหน่งของพื้นที่เก็บข้อมูลระบบ

ข้อสำคัญ
- ตามปกติ เมื่อปิดการแสดงหน้าจอ อย่าใช้บิต **"ปิดหลอดไฟแบ็คไลต์"** ของตำแหน่ง 11 (การควบคุม) โปรดใช้ตำแหน่ง 12 (การเปิด/ปิดการแสดงหน้าจอ)

หมายเหตุ
- **"ตำแหน่งเวิร์ด"** ในตารางนี้คือค่าที่จะปรากฏขึ้น หากคุณทำเครื่องหมายที่ช่อง [Enable System Data Area] และเลือกรายการทั้งหมด

ตำแหน่งเวิร์ด	คำอธิบาย	บิต	รายละเอียด
0	สำรอง	—	สำรอง
1	สถานะ	0 ถึง 1	สำรอง
		2	การพิมพ์
		3	พาร์ทแสดงผลข้อมูล การตั้งค่าการเขียนข้อมูล
		4 ถึง 7	สำรอง
		8	ข้อผิดพลาดในการป้อนข้อมูลในพาร์ทแสดงผลข้อมูล
		9	การเปิด/ปิดการแสดงผล 0:เปิด, 1:ปิด
		10	ตรวจพบว่าหลอดไฟแบ็คไลต์หมดอายุ
		11 ถึง 15	สำรอง
2	สำรอง	—	สำรอง
3	สถานะข้อผิดพลาด	0 ถึง 2	ไม่ใช้งาน
		3	ผลรวมการตรวจสอบหน่วยความจำหน้าจอ
		4	SIO เฟรมมิ่ง
		5	SIO พาริตี้
		6	SIO โอเวอร์รัน
		7 ถึง 9	ไม่ใช้งาน
		10	แบตเตอรี่สำรองมีแรงดันไฟฟ้าต่ำ
		11 ถึง 15	ไม่ใช้งาน
4	ค่า "ปี" ปัจจุบันของนาฬิกา	0 ถึง 7	เลข 2 หลักสุดท้ายของปี (เลข BCD 2 หลัก)
		8 ถึง 15	ไม่ใช้งาน
5	ค่า "เดือน" ปัจจุบันของนาฬิกา	0 ถึง 7	01 ถึง 12 (เลข BCD 2 หลัก)
		8 ถึง 15	ไม่ใช้งาน
6	ค่า "วัน" ปัจจุบันของนาฬิกา	0 ถึง 7	01 ถึง 31 (เลข BCD 2 หลัก)
		8 ถึง 15	ไม่ใช้งาน
7	ค่า "ชั่วโมง" ปัจจุบันของนาฬิกา	0 ถึง 7	00 ถึง 23 (เลข BCD 2 หลัก)
		8 ถึง 15	ไม่ใช้งาน
8	ค่า "นาที" ปัจจุบันของนาฬิกา	0 ถึง 7	00 ถึง 59 (เลข BCD 2 หลัก)
		8 ถึง 15	ไม่ใช้งาน
9	สำรอง	—	สำรอง

ต่อ

ตำแหน่งเวิร์ด	คำอธิบาย	บิต	รายละเอียด
10	การส่งออกสัญญาณอินเตอร์รัพต์ (เมื่อปิดการแตะหน้าจอ)	—	หากเขียนข้อมูลลงในสวิตช์ตั้งค่าตำแหน่งเวิร์ด (16 บิต) เมื่อคุณเอานิ้วออกจากสวิตช์ บิต 8 บิตล่าง จะถูกส่งออกไปเป็นรหัสสัญญาณอินเตอร์รัพต์
11	ตัวแปรควบคุม	0	ปิดหลอดไฟแบ็คไลต์
		1	เปิดออดสัญญาณ
		2	เริ่มการพิมพ์
		3	สำรอง
		4	ออดสัญญาณ
		5	AUX Output
		6	เขียน “FFh” เมื่อคุณแตะหน้าจอและกลับไปที่หน้าจอ (จาก “ปิดการแสดงผล” เป็น “เปิดการแสดงผล”) 0: ไม่ส่งสัญญาณอินเตอร์รัพต์ออกไป 1: ส่งสัญญาณอินเตอร์รัพต์ออกไป
		7 ถึง 10	สำรอง
		11	ยกเลิกการพิมพ์
		12 ถึง 15	สำรอง
12	การเปิด/ปิดการแสดงหน้าจอ	—	ปิดการแสดงหน้าจอด้วย FFFFh แสดงหน้าจอด้วย 0h
13	การส่งออกสัญญาณอินเตอร์รัพต์ (เมื่อเปิดการแตะหน้าจอ)	—	เมื่อเขียนข้อมูลลงในสวิตช์ตั้งค่าตำแหน่งเวิร์ด (16 บิต) บิต 8 บิตล่างจะถูกส่งออกไปเป็นรหัสสัญญาณอินเตอร์รัพต์
14	สำรอง	—	สำรอง
15	หมายเลขหน้าจอที่เปลี่ยนแล้ว		<เมื่อใช้ “หมายเลขหน้าจอที่เปลี่ยนแล้ว” กับอุปกรณ์/PLC> 1 ถึง 9999 (BIN) 1 ถึง 7999 (BCD)
16	การควบคุมหน้าต่าง	0	การแสดงหน้าต่าง 0:ปิด, 1:เปิด
		1	เปลี่ยนลำดับการซ้อนทับของหน้าต่าง 0: เปลี่ยนได้, 1: เปลี่ยนไม่ได้
		12 ถึง 15	สำรอง
17	หมายเลขหน้าต่าง	—	เลขทะเบียนของหน้าต่างแบบแสดงทุกหน้าจอที่เลือกโดยการกำหนดโดยอ้อม 1 ถึง 2000 (BIN/BCD)
18	ตำแหน่งการแสดงหน้าต่าง (พิกัด X)	—	ตำแหน่งการแสดงด้านซ้ายบนของหน้าต่างแบบแสดงทุกหน้าจอที่เลือกโดยการกำหนดโดยอ้อม (BIN/BCD)
19	ตำแหน่งการแสดงหน้าต่าง (พิกัด Y)	—	

<table>
<tr><th>คำอธิบาย</th><th>รายละเอียด</th></tr>
<tr><td>สำรอง</td><td>ตำแหน่ง “0”, “2”, “9” และ “14” จะถูกสำรองไว้
เนื่องจากมีการใช้ตำแหน่งเหล่านี้ภายใน GP โปรดอย่าเขียนข้อมูลลงไปในตำแหน่งเหล่านี้ มิฉะนั้นจะทำงานไม่ถูกต้อง</td></tr>
<tr><td>สถานะ</td><td>โปรดตรวจดูว่าใช้เฉพาะบิตที่จำเป็นเท่านั้น บางครั้ง บิต “สำรอง” จะใช้สำหรับการดูแลรักษาระบบ GP ดังนั้น โปรดอย่าเปิด/ปิดบิตเหล่านี้
<table>
<tr><th>บิต</th><th>คำอธิบาย</th><th>รายละเอียด</th></tr>
<tr><td>0,1</td><td>สำรอง</td><td>-</td></tr>
<tr><td>2</td><td>การพิมพ์</td><td>บิตนี้จะเปิดขึ้นระหว่างพิมพ์ข้อมูล ขณะบิตนี้เปิด ข้อมูลส่งออกอาจเสียหายได้หากหน้าจอเปลี่ยนเป็นหน้าจอแบบออฟไลน์</td></tr>
<tr><td>3</td><td>การตั้งค่าการเขียนข้อมูล</td><td>บิตนี้จะถูกเปลี่ยนเป็นตรงกันข้ามทุกครั้งที่มีการเขียนข้อมูลจากพาร์ทแสดงผลข้อมูล (การป้อนข้อมูลการตั้งค่า)</td></tr>
<tr><td>4 ถึง 7</td><td>สำรอง</td><td>-</td></tr>
<tr><td>8</td><td>ข้อผิดพลาดในการป้อนข้อมูลลงในพาร์ทแสดงผลข้อมูล</td><td>เมื่อคุณป้อนข้อมูลลงในพาร์ทแสดงผลข้อมูลที่ตั้งค่าการแจ้งเตือนไว้ โดยป้อนค่าที่อยู่นอกช่วงการแจ้งเตือน บิตนี้จะเปิดขึ้น
เมื่อคุณป้อนค่าที่อยู่ภายในช่วงการแจ้งเตือนหรือเปลี่ยนหน้าจอ บิตนี้จะปิด</td></tr>
<tr><td>9</td><td>การเปิด/ปิดการแสดงผล
(0: เปิด, 1:ปิด)</td><td>บิตนี้สามารถตรวจว่าจะเปิด/ปิดการแสดงหน้าจอของ GP จากอุปกรณ์/PLC หรือไม่

บิตนี้จะเปลี่ยนไปในกรณีต่อไปนี้
(1) เมื่อมีการเขียนค่า FFFFh ลงในบิตเปิด/ปิดการแสดงผลของพื้นที่เก็บข้อมูลระบบ การแสดงผลจะปิด
(2) เมื่อครบระยะเวลาแสตนด์บายที่กำหนดไว้ การแสดงผลจะปิดโดยอัตโนมัติ
(3) หากหน้าจอเปลี่ยนไปหรือถูกแตะหลังจากการแสดงผลปิดลง การแสดงผลจะกลับมาเปิดใหม่

หมายเหตุ
• บิตนี้ไม่สามารถเปลี่ยนบิต 0 ของตำแหน่ง LS0014 “การควบคุม” (ปิดหลอดไฟแบ็คไลต์)</td></tr>
<tr><td>10</td><td>ตรวจพบว่าหลอดไฟแบ็คไลต์หมดอายุ</td><td>เมื่อตรวจพบว่าหลอดไฟแบ็คไลต์หมดอายุ บิตนี้จะเปิดขึ้น</td></tr>
<tr><td>11 ถึง 15</td><td>สำรอง</td><td>-</td></tr>
</table>
</td></tr>
</table>

ต่อ

<table>
<tr><th>คำอธิบาย</th><th>รายละเอียด</th></tr>
<tr><td>สถานะข้อผิดพลาด</td><td>
เมื่อเกิดข้อผิดพลาดใน GP บิตที่เกี่ยวข้องจะเปิดขึ้น หลังจากบิตเปิดขึ้นและเครื่องดับลง สถานะจะคงอยู่จนกว่าจะเปลี่ยนจากโหมดออฟไลน์เป็นโหมดแอ็คทีฟ
<table>
<tr><th>บิต</th><th>คำอธิบาย</th><th>รายละเอียด</th></tr>
<tr><td>0 ถึง 2</td><td>ไม่ใช้งาน</td><td></td></tr>
<tr><td>3</td><td>ผลรวมการตรวจสอบ
หน่วยความจำหน้าจอ</td><td>มีข้อผิดพลาดในไฟล์โปรเจค
โปรดถ่ายโอนไฟล์อีกครั้ง</td></tr>
<tr><td>4</td><td>SIO เฟรมมิ่ง</td><td></td></tr>
<tr><td>5</td><td>SIO พาริตี้</td><td></td></tr>
<tr><td>6</td><td>SIO โอเวอร์รัน</td><td></td></tr>
<tr><td>7 ถึง 9</td><td>ไม่ใช้งาน</td><td></td></tr>
<tr><td>10</td><td>แบตเตอรี่สำรองมีแรงดัน
ไฟฟ้าต่ำ</td><td>บิตนี้จะเปิดขึ้นเมื่อแบตเตอรี่ลิเธียมสำรองมีแรงดันไฟฟ้าเหลือน้อย แบตเตอรี่สำรองถูกใช้โดยนาฬิกาและหน่วยความจำสำรองข้อมูล</td></tr>
<tr><td>11 ถึง 15</td><td>ไม่ใช้งาน</td><td></td></tr>
</table>
เนื่องจากตำแหน่งเหล่านี้ใช้สำหรับการควบคุมระบบ โปรดอย่าให้พาร์ทแสดงผลข้อมูลแสดงตำแหน่งเหล่านี้
</td></tr>
<tr><td>ข้อมูลนาฬิกา (ปัจจุบัน)</td><td>
ไม่ว่าจะเป็นค่าอะไรก็ตาม ระบบจะจัดเก็บค่าไว้เป็นชนิด BCD ในบิตลำดับสูงสุด บิต 0 ถึงบิต 7
[Year] คือเลข 2 หลักสุดท้ายของปี, [Month] คือเลข 2 หลักตั้งแต่ 01 ถึง 12, [Day] คือเลข 2 หลักตั้งแต่ 01 ถึง 31, [Hour] คือเลขชั่วโมง 2 หลักตั้งแต่ 00 ถึง 23 และ [Minute] คือเลขนาที 2 หลักตั้งแต่ 00 ถึง 59
■ตัวอย่างการตั้งค่า: <October 19th, 2005, 21:57>
• “ปี” – เขียน “0005” ลงในตำแหน่งเวิร์ด “4”
• “เดือน” – เขียน “0010” ลงในตำแหน่งเวิร์ด “5”
• “วัน” – เขียน “0019” ลงในตำแหน่งเวิร์ด “6”
• “ชั่วโมง” – เขียน “0021” ลงในตำแหน่งเวิร์ด “7”
• “นาที” – เขียน “0057” ลงในตำแหน่งเวิร์ด “8”
</td></tr>
<tr><td>การส่งออกสัญญาณอินเตอร์รัพต์ (เมื่อปิดการแตะหน้าจอ)</td><td>
หากเขียนข้อมูลลงในสวิตช์ตั้งค่าตำแหน่งเวิร์ด (16 บิต) เมื่อคุณเอานิ้วออกจากสวิตช์ บิต 8 บิตล่างจะถูกส่งออกไปเป็นรหัสสัญญาณอินเตอร์รัพต์ (จะไม่ส่งรหัสควบคุม “FFh” ออกไป)
โปรดอย่าเขียนรหัสควบคุมในช่วง 00 ถึง 1F เพราะอาจทำให้เกิดปัญหาในการสื่อสาร
</td></tr>
</table>

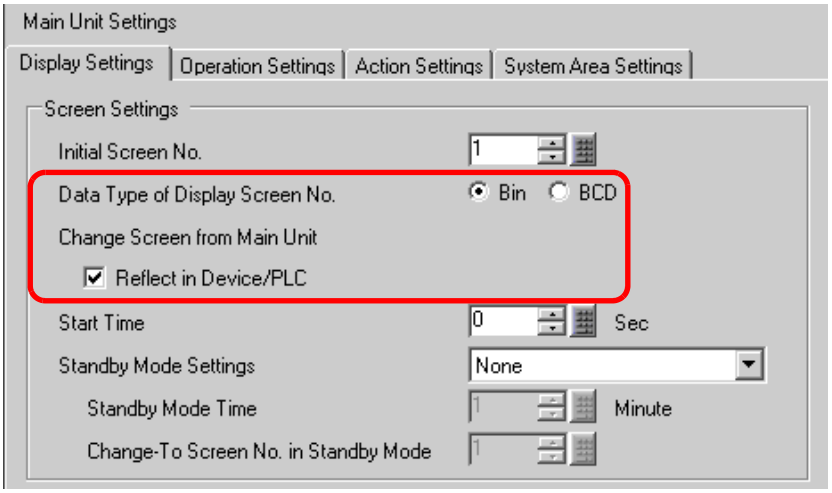
ต่อ

<table>
<tr><th>คำอธิบาย</th><th>รายละเอียด</th></tr>
<tr><td>ตัวแปรควบคุม</td><td>
หมายเหตุ
• โปรดตรวจสอบให้แน่ใจว่าได้เขียนตำแหน่งนี้ในหน่วยบิต ในบางกรณี การเขียนด้วยข้อมูลเวิร์ดจะทำให้ค่าเปลี่ยนไป
• บางครั้ง บิต “สำรอง” จะใช้สำหรับการดูแลรักษาระบบ GP ดังนั้น โปรดปิดบิตเหล่านี้
<table>
<tr><th>บิต</th><th>คำอธิบาย</th><th>รายละเอียด</th></tr>
<tr><td>0</td><td>ปิดหลอดไฟแบ็คไลต์</td><td>เมื่อบิตนี้เปิด หลอดไฟแบ็คไลต์จะปิด เมื่อบิตนี้ปิด หลอดไฟแบ็คไลต์จะเปิด (พาร์ทที่วางอยู่บนหน้าจอจะยังคงทำงานหลังจากจอ LCD สว่างขึ้น)
หมายเหตุ
• โดยทั่วไปแล้ว โปรดใช้ตำแหน่งเวิร์ด “12” (การเปิด/ปิดการแสดงหน้าจอ) ในการปิดการแสดงหน้าจอ</td></tr>
<tr><td>1</td><td>เปิดออดสัญญาณ</td><td>0: ไม่ส่งข้อมูลออก, 1: ส่งข้อมูลออก</td></tr>
<tr><td>2</td><td>เริ่มการพิมพ์</td><td>0: ไม่ส่งข้อมูลออก, 1: ส่งข้อมูลออก
เมื่อบิตนี้เปิดขึ้น การคัดลอกหน้าจอจะเริ่มขึ้น
หมายเหตุ
• เมื่อ “บิต 2” สถานะ (การพิมพ์) เปิดขึ้น โปรดปิดบิตด้วยตนเอง</td></tr>
<tr><td>3</td><td>สำรอง</td><td>0: กำหนดไว้ตายตัว</td></tr>
<tr><td>4</td><td>ออดสัญญาณ</td><td>การดำเนินการต่อไปนี้เกิดขึ้นเฉพาะเมื่อ “บิต 1” การควบคุม (เปิดออดสัญญาณ) เปิดขึ้นเท่านั้น
0: ส่งข้อมูลออก, 1: ไม่ส่งข้อมูลออก
หากต้องการหยุดเสียงออดสัญญาณ ให้ปิดบิตนี้</td></tr>
<tr><td>5</td><td>AUX Output</td><td>การดำเนินการต่อไปนี้เกิดขึ้นเฉพาะเมื่อ “บิต 1” การควบคุม (เปิดออดสัญญาณ) เปิดขึ้นเท่านั้น
0: ส่งข้อมูลออก, 1: ไม่ส่งข้อมูลออก
หากต้องการหยุดการส่งสัญญาณเสียงออกผ่านขั้วต่อ AUX ให้เปิดบิตนี้</td></tr>
<tr><td>6 ถึง10</td><td>สำรอง</td><td>0: กำหนดไว้ตายตัว</td></tr>
<tr><td>11</td><td>ยกเลิกการพิมพ์</td><td>0: ส่งข้อมูลออก, 1: ไม่ส่งข้อมูลออก
เมื่อบิตเปิด การพิมพ์ในปัจจุบันทั้งหมดจะถูกยกเลิก
หมายเหตุ
• หลังจากการพิมพ์หยุดลง เมื่อ “บิต 2” สถานะ (การพิมพ์) ปิดลง โปรดปิดบิตด้วยตนเอง
• ถึงแม้บิตยกเลิกการพิมพ์จะเปิดขึ้น แต่เครื่องพิมพ์จะยังคงพิมพ์ข้อมูลในหน่วยความจำที่ได้รับมา</td></tr>
<tr><td>12 ถึง 15</td><td>(สำรอง)</td><td>0: กำหนดไว้ตายตัว</td></tr>
</table>
</td></tr>
<tr><td>การเปิด/ปิด
การแสดงหน้าจอ</td><td>
แสดงหน้าจอเมื่อค่าเป็น “0h” ซ่อนหน้าจอเมื่อค่าเป็น “FFFFh” ค่าอื่นที่ไม่ใช่ “0h” และ “FFFFh” เป็นค่าที่สำรองไว้ เมื่อหน้าจอถูกซ่อน (เมื่อค่ากลายเป็น “FFFFh”) เมื่อแตะหน้าจอในครั้งต่อไปจะทำให้หน้าจอกลับมาแสดงผลอีก
⃠ เนื่องจากตำแหน่งเหล่านี้ใช้สำหรับการควบคุมระบบ โปรดอย่าให้พาร์ทแสดงผลข้อมูลแสดงตำแหน่งเหล่านี้
⃠ เนื่องจากตำแหน่งถูกควบคุมในหน่วยเวิร์ด คุณจึงเขียนบิตไม่ได้
⃠ เมื่อคุณเขียน “FFFFh” หน้าจอที่แสดงอยู่จะหายไปชั่วคราว หากคุณต้องการให้การแสดงหน้าจอหายไปเป็นระยะเวลาเท่ากับระยะเวลาของโหมดแสตนด์บายที่กำหนดไว้ในการตั้งค่าเริ่มต้นของโหมดออฟไลน์ของ GP โปรดเขียน “0000h” ลงไป
</td></tr>
</table>

ต่อ

<table>
<tr><th>คำอธิบาย</th><th>รายละเอียด</th></tr>
<tr><td>การส่งออกสัญญาณอินเตอร์รัพต์
(เมื่อเปิดการแตะหน้าจอ)</td><td>
เมื่อเขียนข้อมูลลงในสวิตช์ตั้งค่าตำแหน่งเวิร์ด (16 บิต) บิต 8 บิตล่างจะถูกส่งออกจาก GP ไปยังโฮสต์เป็นรหัสสัญญาณอินเตอร์รัพต์
โปรดอย่าเขียนรหัสควบคุมในช่วง 00 ถึง 1F เพราะอาจทำให้เกิดปัญหาในการสื่อสาร
เนื่องจากตำแหน่งเหล่านี้ใช้สำหรับการควบคุมระบบ โปรดอย่าให้พาร์ทแสดงผลข้อมูลแสดงตำแหน่งเหล่านี้
เนื่องจากตำแหน่งถูกควบคุมในหน่วยเวิร์ด คุณจึงเขียนบิตไม่ได้

หมายเหตุ
• เมื่อคุณเขียนข้อมูลด้วยสวิตช์ตั้งค่าตำแหน่งเวิร์ด (16 บิต) ข้อมูลจะถูกส่งออกไปเป็นข้อมูลอินเตอร์รัพต์ คุณสามารถค้นไบต์ของการนำเข้าสัญญาณอินเตอร์รัพต์ในโฮสต์ (ด้วยคำสั่ง INPUT$ ในภาษา BASIC เป็นต้น) และปรับโปรแกรมให้ง่ายขึ้นโดยใช้การส่งออกสัญญาณอินเตอร์รัพต์ที่ค้นได้ เพื่อข้ามไปยังแต่ละรูทีนย่อย
</td></tr>
<tr><td>หมายเลขหน้าจอที่เปลี่ยนแล้ว</td><td>
ตั้งค่าหมายเลขหน้าจอที่เปลี่ยนแล้ว ช่วงการตั้งค่าจะแตกต่างกันไป ขึ้นอยู่กับว่าตั้งค่า [Data Type of Display Screen No.] และ [Change Screen from Main Unit - Reflect in Device/PLC] บนการตั้งค่าระบบ - [Main Unit Settings] - แท็บ [Display Settings] ไว้หรือไม่



เมื่อตั้งค่า [Data Type of Display Screen No.] เป็น [Bin]
<table>
<tr><th>Reflect in Device/ PLC Screen</th><th>เปลี่ยนจากอุปกรณ์/ หน้าจอ PLC</th><th>เปลี่ยนจากยูนิตหลัก (สวิตช์ เป็นต้น)</th></tr>
<tr><td>เลือก</td><td>1 ถึง 9999</td><td>1 ถึง 9999</td></tr>
<tr><td>ไม่เลือก</td><td>1 ถึง 9999</td><td>1 ถึง 9999</td></tr>
</table>
เมื่อตั้งค่า [Data Type of Display Screen No.] เป็น [BCD]
<table>
<tr><th>Reflect in Device/ PLC Screen</th><th>เปลี่ยนจากอุปกรณ์/ หน้าจอ PLC</th><th>เปลี่ยนจากยูนิตหลัก (สวิตช์ เป็นต้น)</th></tr>
<tr><td>เลือก</td><td>1 ถึง 7999</td><td>1 ถึง 7999</td></tr>
<tr><td>ไม่เลือก</td><td>1 ถึง 7999</td><td>1 ถึง 7999</td></tr>
</table>
เนื่องจากตำแหน่งเหล่านี้ใช้สำหรับการควบคุมระบบ โปรดอย่าให้พาร์ทแสดงผลข้อมูลแสดงตำแหน่งเหล่านี้
เนื่องจากตำแหน่งถูกควบคุมในหน่วยเวิร์ด คุณจึงเขียนบิตไม่ได้
</td></tr>
</table>

ต่อ

คำอธิบาย	รายละเอียด
การควบคุมหน้าต่าง	ควบคุมการแสดงหน้าต่าง ☞ คู่มืออ้างอิงสำหรับ GP-Pro EX “18.7.2 Word Operation“ (หน้า 18-23)
หมายเลขหน้าต่าง	จัดเก็บเลขทะเบียนของหน้าต่างแบบแสดงทุกหน้าจอที่เลือกโดยการกำหนดโดยอ้อม ตั้งแต่ 1 ถึง 2000 (BIN/BCD)
ตำแหน่งการแสดงหน้าต่าง	จัดเก็บตำแหน่งการแสดงด้านซ้ายบนของหน้าต่างแบบแสดงทุกหน้าจอที่เลือกโดยการกำหนดโดยอ้อม “+18” แสดงถึงพิกัด X, “+19” แสดงถึงพิกัด Y ข้อมูลเป็นชนิด BIN หรือ BCD

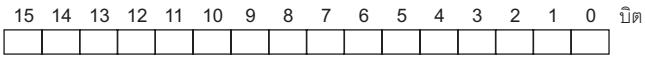
1.1.3 รีเลย์พิเศษ

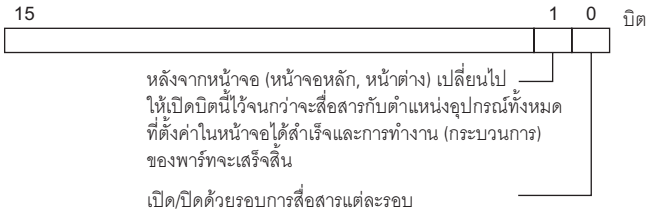
รีเลย์พิเศษไม่ได้ป้องกันการเขียนไว้ ดังนั้น อย่าเปิด/ปิดด้วยพาร์ทหรือเวิร์ดเขียนข้อมูล

รีเลย์พิเศษมีโครงสร้างดังต่อไปนี้

วิธีการเชื่อมต่อผ่านหน่วยความจำ

ตำแหน่ง	คำอธิบาย
2032	ข้อมูลรีเลย์ร่วม
2033	ข้อมูลหน้าจอหลัก
2034	สำรอง
2035	ตัวนับเลขฐานสอง 1 วินาที
2036	เวลาสำหรับการแสดงผล
2037	สำรอง
2038	ตัวนับเวลาสำหรับการแสดงผล
2039	สำรอง
2040	สำรอง
2041	
2042	
2043	
2044	
2045	
2046	
2047	

<table>
<tr><th>คำอธิบาย</th><th>รายละเอียด</th></tr>
<tr><td>ข้อมูลรีเลย์ร่วม (2032)</td><td>


<table>
<tr><th>บิต</th><th>คำอธิบาย</th></tr>
<tr><td>0</td><td>สำรอง</td></tr>
<tr><td>1</td><td>หลังจากหน้าจอ (หน้าจอหลัก, หน้าจอหน้าต่าง) เปลี่ยนไป บิตนี้จะเปิดจนกว่าจะจัดการพาร์ทเสร็จสมบูรณ์</td></tr>
<tr><td>2</td><td>สำรอง</td></tr>
<tr><td>3</td><td>เปิดขณะหน้าจอเริ่มต้นปรากฏขึ้นเมื่อเปิดเครื่อง</td></tr>
<tr><td>4</td><td>ตามปกติจะเปิด</td></tr>
<tr><td>5</td><td>ตามปกติจะปิด</td></tr>
<tr><td>6</td><td>เปิดเมื่อมีการลบข้อมูลในหน่วยความจำสำรองข้อมูลออก
(เฉพาะหน่วยความจำสำรองข้อมูลบนบอร์ดไมโครคอมพิวเตอร์เท่านั้น)</td></tr>
<tr><td>7</td><td>ขณะใช้ D-Scripts อยู่ บิตนี้จะเปิดขึ้นเมื่อเกิดข้อผิดพลาดของ BCD</td></tr>
<tr><td>8</td><td>ขณะใช้ D-Scripts อยู่ บิตนี้จะเปิดขึ้นเมื่อเกิดข้อผิดพลาดที่เป็นค่าศูนย์</td></tr>
<tr><td>9</td><td>เปิดเมื่อไม่สามารถถ่ายโอนข้อมูลไฟล์ลิ่งฟังก์ชันไปยังหน่วยความจำสำรองข้อมูล</td></tr>
<tr><td>10</td><td>เปิดเมื่อข้อมูลไฟล์ลิ่งฟังก์ชันที่ถูกถ่ายโอนตามตำแหน่งเวิร์ดควบคุมนั้นไม่สามารถถ่ายโอนจาก PLC*1 ไปยัง SRAM ได้
นอกจากนี้ หากมีการถ่ายโอนข้อมูลระหว่าง PLC ด้วยพาร์ทแสดงผลข้อมูลพิเศษ เมื่อมีตำแหน่งบิตเสร็จสิ้นสถานะการถ่ายโอน บิตนี้จะเปิดขึ้นเมื่อไม่สามารถถ่ายโอนข้อมูลจาก PLC*1 ไปยังพื้นที่ หรือจาก PLC*1 ไปยัง SRAM</td></tr>
<tr><td>11</td><td>เปิดขณะถ่ายโอนข้อมูลไฟล์ลิ่งฟังก์ชันระหว่าง
SRAM และพื้นที่ LS*1 ด้วยพาร์ทแสดงผลข้อมูลพิเศษ [Filing]</td></tr>
<tr><td>12</td><td>เมื่อใช้ D-Scripts บิตนี้จะเปิดเมื่อเกิดข้อผิดพลาดในการสื่อสารจากคำสั่ง memcpy()
หรือการอ่านข้อมูลการกำหนดค่าออฟเซ็ตของตำแหน่ง และปิดเมื่อเสร็จสิ้นการอ่านข้อมูลตามปกติ</td></tr>
<tr><td>13</td><td>ใน [System Settings] - [Script Settings] เมื่อไม่ได้ตั้งค่า [D-Script/Global DScript]
ในโปรเจค บิตนี้จะเปิดเมื่อมีการอ่านข้อมูลของฟังก์ชันส่ง, ฟังก์ชันรับ, ตัวแปรควบคุม,
ตัวแปรสถานะ และจำนวนข้อมูลที่ได้รับในการตั้งค่าป้ายชื่อของ [SIO Port Operation]</td></tr>
<tr><td>14</td><td>ใน [System Settings] - [Script Settings] เมื่อตั้งค่า [D-Script/Global D-Script]
ในโปรเจคไว้ บิตนี้จะเปิดเมื่อมีการเรียกใช้ฟังก์ชัน [Text Operation] ของ extended script
นอกจากนี้ ใน [System Settings] - [Script Settings] เมื่อตั้งค่า [Extended Script]
ในโปรเจคไว้ บิตนี้จะเปิดแม้ว่าจะมีการเรียกใช้ฟังก์ชัน I/O (IO_WRITE, IO_READ)
ของ D-Script/Global D-Script [SIO Port Operation] ก็ตาม</td></tr>
<tr><td>15</td><td>สำรอง</td></tr>
</table>
*1 สำหรับวิธีการเชื่อมต่อผ่านหน่วยความจำ หมายถึง “พื้นที่สำหรับผู้ใช้” ภายในพื้นที่ระบบ
</td></tr>
<tr><td>ข้อมูลหน้าจอหลัก (2033)</td><td>


</td></tr>
<tr><td>สำรอง
(2034, 2037, 2040 ถึง 2047)</td><td>ยังไม่ได้กำหนดค่าของตำแหน่งสำรอง โปรดอย่าใช้ตำแหน่งเหล่านี้</td></tr>
</table>

ต่อ

คำอธิบาย	รายละเอียด
ตัวนับเลขฐานสอง 1 วินาที (2035)	เพิ่มขึ้นครั้งละหนึ่งวินาทีทันทีหลังจากเปิดเครื่อง ข้อมูลเป็นเลขฐานสอง
เวลาสำหรับการแสดงผล (2036)	เวลาที่ใช้ในการแสดงผลโดยเริ่มตั้งแต่พาร์ทแรกสุดที่ตั้งค่าบนหน้าจอแสดงผลไปจนถึงตอนสิ้นสุดของพาร์ทสุดท้าย ข้อมูลจะจัดเก็บในรูปแบบเลขฐานสองโดยมีหน่วยเป็นมิลลิวินาที เมื่อเสร็จสิ้นการประมวลผลล่วงหน้าของพาร์ทเป้าหมายแล้ว ข้อมูลจะได้รับการอัพเดต ค่าเริ่มต้นของข้อมูลคือ “0” และมีความคลาดเคลื่อนเท่ากับ ±10 มิลลิวินาที
ตัวนับเวลาสำหรับการแสดงผล (2038)	ตัวนับจะเพิ่มขึ้นทุกครั้งที่พาร์ทซึ่งตั้งค่าบนหน้าจอแสดงผลทำงาน ข้อมูลเป็นเลขฐานสอง

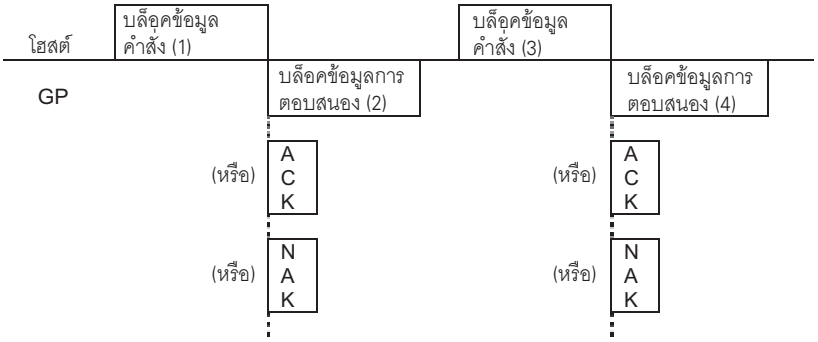
2 คำสั่งของการเชื่อมต่อผ่านหน่วยความจำ

2.1 การควบคุมโปรโตคอลการสื่อสารขั้นพื้นฐาน

กระบวนการพื้นฐานในการควบคุมโปรโตคอลการสื่อสารมีดังต่อไปนี้:

2.1.1 SIO

■ **การถ่ายโอนข้อมูลจากโฮสต์ไปที่** GP



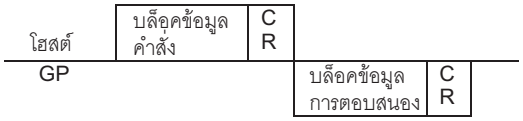
- (1) และ (3) (พื้นที่เก็บข้อมูลคำสั่ง) ทำหน้าที่จัดเก็บข้อมูลที่จะส่งจากอุปกรณ์โฮสต์ไปยัง GP
- หลังจากที่ GP วิเคราะห์ข้อมูลคำสั่งแล้ว (2) และ (4) (พื้นที่เก็บข้อมูลการตอบสนอง) จะจัดเก็บผลลัพธ์ของ “ACK“ หรือ “NAK“ หรือไม่มีการตอบสนอง
- โปรดส่งข้อมูลคำสั่ง (3) จากอุปกรณ์โฮสต์ หลังจากได้รับข้อมูลตอบสนอง (2) จาก GP แล้ว

■ **การถ่ายโอนข้อมูลจาก** GP **ไปที่โฮสต์**

- เมื่อป้อนข้อมูลด้วยหน้าจอสัมผัส GP จะส่งข้อมูลที่จัดเก็บในพื้นที่เก็บข้อมูลอินเตอร์รัพต์ไปที่อุปกรณ์โฮสต์ (การส่งออกสัญญาณอินเตอร์รัพต์)
- จะไม่มีการส่งออกสัญญาณอินเตอร์รัพต์ หากเป็นการเชื่อมต่อแบบ 1:n หรือ RS422/485 (2wire) โปรดดูที่การร้องขอส่งออกสัญญาณอินเตอร์รัพต์

2.1.2 การสื่อสารในโหมดการแปลง SIO

■ **การถ่ายโอนข้อมูลจากโฮสต์ไปที่** GP

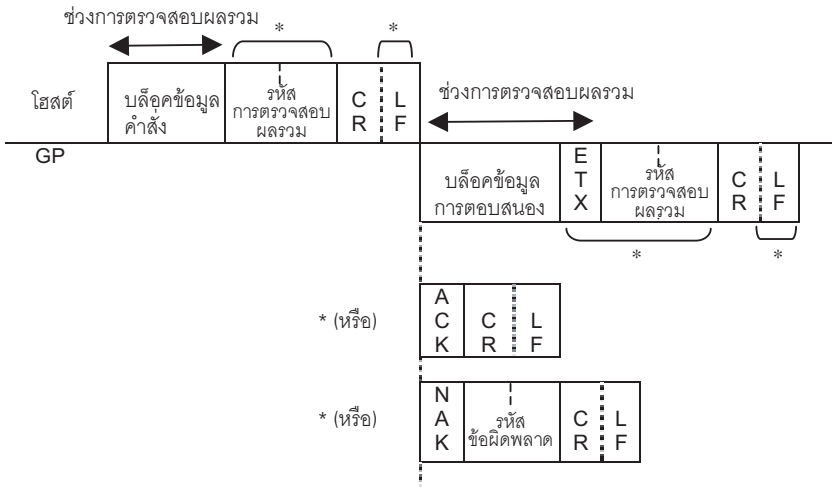


■ **การถ่ายโอนข้อมูลจาก** GP **ไปที่โฮสต์ (การส่งออกสัญญาณอินเตอร์รัพต์)**

- คุณไม่สามารถส่งออกสัญญาณอินเตอร์รัพต์ได้หาก SIO เป็นชนิด RS244 (2wire)

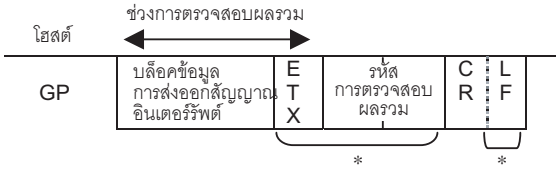
2.1.3 การสื่อสารในโหมดขยาย SIO (1:1 ASCII)

■ **การถ่ายโอนข้อมูลจากโฮสต์ไปที่** GP



- อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

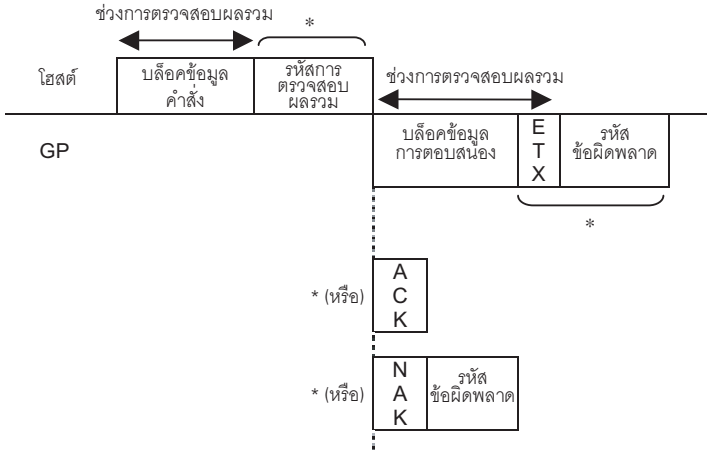
■ การถ่ายโอนข้อมูลจาก GP ไปที่โฮสต์ (การส่งออกสัญญาณอินเตอร์รัพต์)



- อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
- ในกรณีที่เป็น SIO ชนิด RS244/485 (2wire) หรือการเชื่อมต่อแบบ UDP โปรดใช้ “คำสั่งร้องขอส่งออกสัญญาณอินเตอร์รัพต์” ในการส่งออกสัญญาณอินเตอร์รัพต์

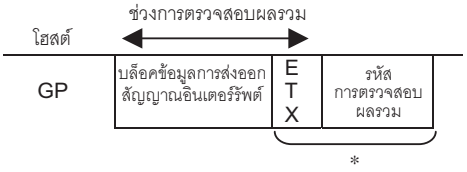
2.1.4 การสื่อสารในโหมดขยาย SIO (1:1 Binary)

■ การถ่ายโอนข้อมูลจากโฮสต์ไปที่ GP



- อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

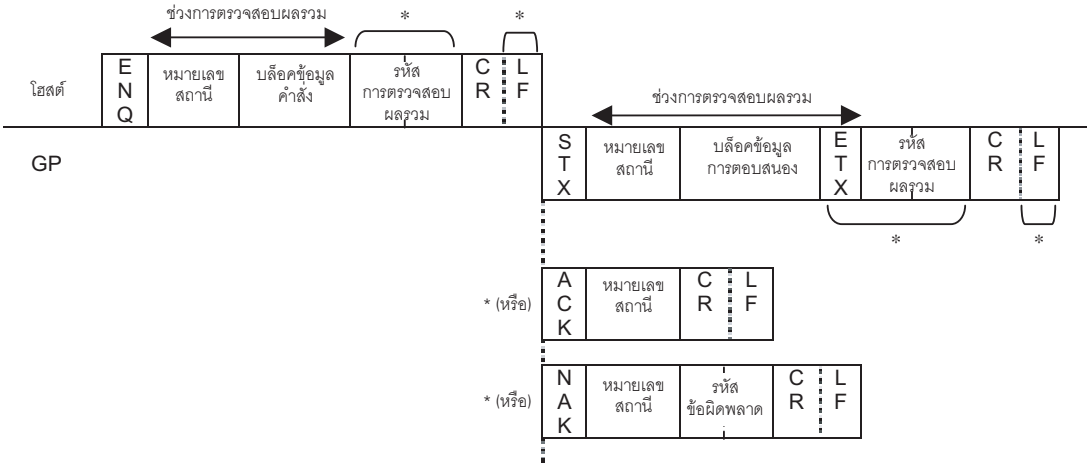
■ การถ่ายโอนข้อมูลจาก GP ไปที่โฮสต์ (การส่งออกสัญญาณอินเตอร์รัพต์)



- อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
- คุณไม่สามารถใช้การควบคุม XON/XOFF ในโหมดเลขฐานสองได้ ให้ใช้การควบคุม ER และเปิดใช้การตอบสนอง (ACK/NAK) ในการสื่อสาร
- ในกรณีที่เป็น SIO ชนิด RS244/485 (2wire) โปรดใช้ “คำสั่งร้องขอส่งออกสัญญาณอินเตอร์รัพต์” ในการส่งออกสัญญาณอินเตอร์รัพต์

2.1.5 การสื่อสารในโหมดขยาย SIO (1:n ASCII)

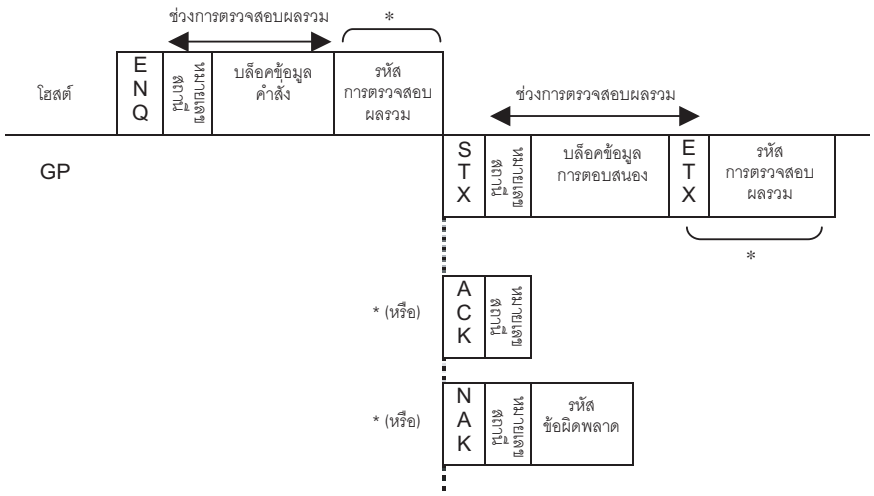
■ การถ่ายโอนข้อมูลจากโฮสต์ไปที่ GP



- อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
- คุณสามารถตั้งค่าหมายเลขสถานีเป็น “FF“ เพื่อถ่ายโอนคำสั่งไปยังทุกสถานีพร้อมกัน โปรดทราบว่าจะไม่มีการตอบสนอง ACK หรือ NAK ในกรณีเช่นนี้ ให้ตั้งค่าช่วงเวลาที่จะส่งคำสั่งถัดไปหลังจากที่ส่งคำสั่งแรกแล้วไม่น้อยกว่า 100 มิลลิวินาที
  นอกจากนี้ โปรดทราบว่าคุณไม่สามารถใช้คำสั่ง “อ่านจากพื้นที่ระบบ” (ESC R) หรือ “ค่าปัจจุบันของความสว่าง/คอนทราสต์“ (ESC $) ซึ่งจำเป็นต้องใช้ข้อมูลการตอบสนองดังกล่าว
- หากเป็นการเชื่อมต่อแบบ 1:n โปรดใช้ “คำสั่งร้องขอส่งออกสัญญาณอินเตอร์รัพต์” ในการส่งออกสัญญาณอินเตอร์รัพต์

2.1.6 การสื่อสารในโหมดขยาย SIO (1:n Binary)

■ **การถ่ายโอนข้อมูลจากโฮสต์ไปที่** GP



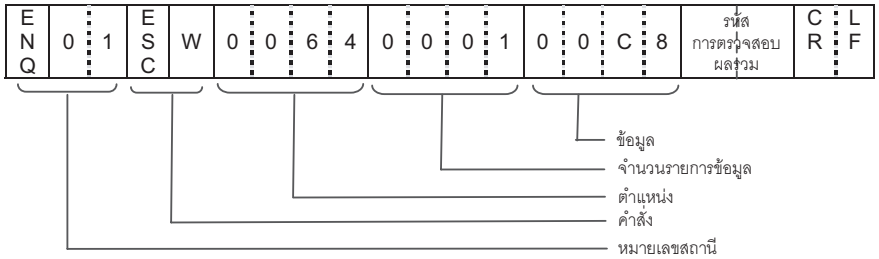
- อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
- คุณสามารถตั้งค่าหมายเลขสถานีเป็น “FF” เพื่อถ่ายโอนคำสั่งไปยังทุกสถานีพร้อมกัน โปรดทราบว่าจะไม่มีการตอบสนอง ACK หรือ NAK ในกรณีเช่นนี้ ให้ตั้งค่าช่วงเวลาที่จะส่งคำสั่งถัดไปหลังจากที่ส่งคำสั่งแรกแล้วไม่น้อยกว่า 100 มิลลิวินาที
  นอกจากนี้ โปรดทราบว่าคุณไม่สามารถใช้คำสั่ง “อ่านจากพื้นที่ระบบ” (ESC R) หรือ “ค่าปัจจุบันของความสว่าง/คอนทราสต์” (ESC $) ซึ่งจำเป็นต้องใช้ข้อมูลการตอบสนองดังกล่าว
- คุณไม่สามารถใช้การควบคุม XON/XOFF ในโหมดเลขฐานสองได้ ให้ใช้การควบคุม ER และเปิดใช้การตอบสนอง (ACK/NAK) ในการสื่อสาร
- หากเป็นการเชื่อมต่อแบบ 1:n โปรดใช้ “คำสั่งร้องขอส่งออกสัญญาณอินเตอร์รัพต์” ในการส่งออกสัญญาณอินเตอร์รัพต์

2.1.7 รหัสการตรวจสอบผลรวม

รหัสการตรวจสอบผลรวมคือไบต์หนึ่งไบต์ล่าง (8 บิต) ของผลรวมของข้อมูลทั้งหมดที่อยู่ในช่วงของการตรวจสอบผลรวม

ในโหมด ASCII ข้อมูลจะถูกแปลงเป็นรหัส ASCII ก่อนทำการหาผลรวม จากนั้น จึงนำตัวเลข 2 หลักล่างของผลรวมเลขฐานสิบหกของข้อมูลทั้งหมดมาใช้เป็นรหัสการตรวจสอบผลรวม
ในโหมดเลขฐานสอง จะนำไบต์ล่างของผลรวมของข้อมูลทั้งหมดมาใช้เป็นรหัสการตรวจสอบผลรวม

ตัวอย่างเช่น: โหมดขยาย, 1:n ASCII
บล็อคข้อมูลดังต่อไปนี้จะเขียนค่า “200” (เลขฐานสิบ) ลงในตำแหน่ง 100 ในพื้นที่ระบบ ดังนี้

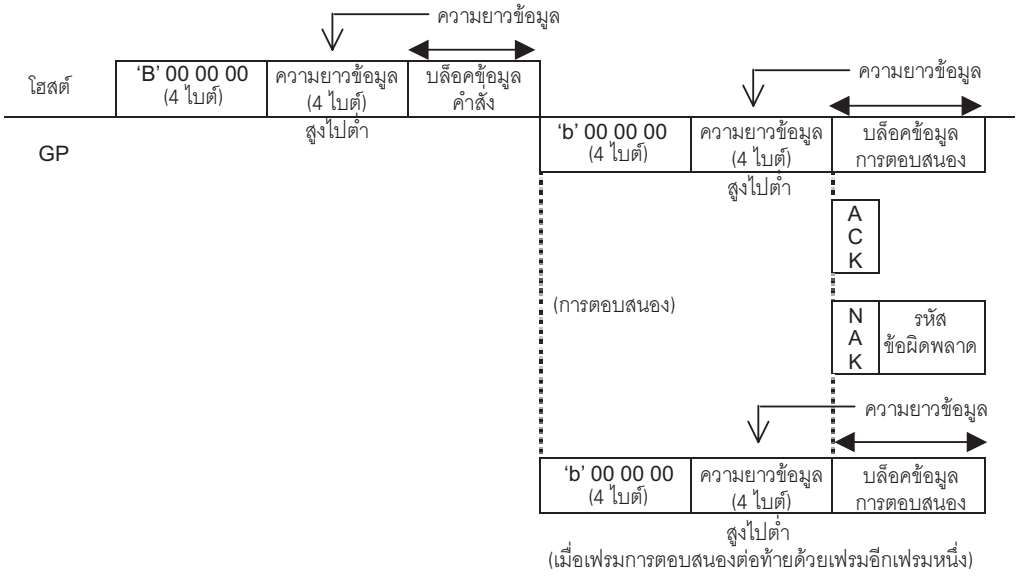


ASCII 30H (0) + 31H (1) + 1BH (ESC) + 57H (W) + 30H (0) + 30H (0) + 36H (6) + 34H (4) + 30H (0) + 30H (0) +
ASCII 30H (0) + 31H (1) + 30H (0) + 30H (0) + 43H (C) + 38H (8)
= 339H

ตัวเลข 2 หลักล่างได้แก่ “39” (33H, 39H) จะถูกใช้เป็นรหัสการตรวจสอบผลรวม

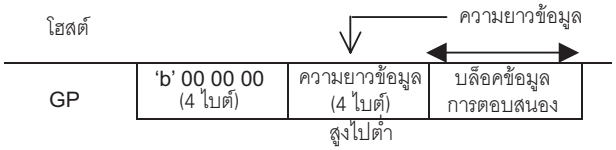
2.1.8 LAN

■ การถ่ายโอนข้อมูลจากโฮสต์ไปที่ GP



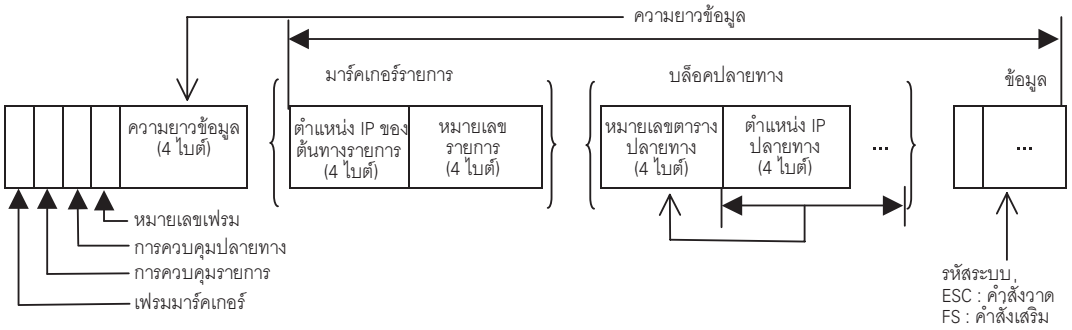
- **พื้นที่เก็บข้อมูลคำสั่งจะจัดเก็บข้อมูลที่จะถูกส่งจากอุปกรณ์โฮสต์ไปยัง GP**
- **หลังจาก GP ทำการวิเคราะห์ข้อมูลคำสั่งแล้ว พื้นที่เก็บข้อมูลตอบสนองจะจัดเก็บผลลัพธ์ระหว่าง "ACK" หรือ "NAK" หรือไม่มีการตอบสนอง**

■ การถ่ายโอนข้อมูลจาก GP ไปที่โฮสต์



■ รายละเอียดเกี่ยวกับรูปแบบเฟรม

เฟรมของ LAN แบบเชื่อมต่อผ่านหน่วยความจำมีโครงสร้างดังนี้:

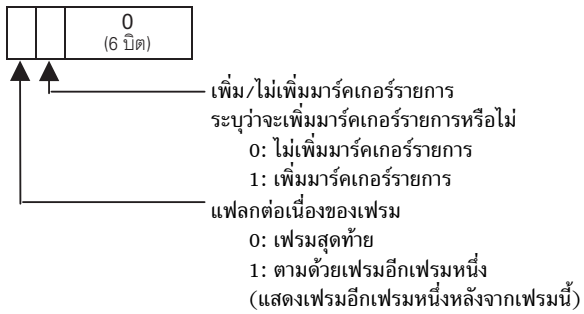


ไบต์ 8 ไบต์แรกตั้งแต่เฟรมมาร์คเกอร์จนถึงความยาวข้อมูลจะถูกจัดเตรียมไว้ในเฟรมของ LAN แบบเชื่อมต่อผ่านหน่วยความจำทุกเฟรม
ดังนั้น ในระหว่างการตรวจสอบเฟรม ระบบจะตรวจสอบไบต์ 8 ไบต์แรกก่อน จากนั้นจึงตรวจสอบข้อมูลถัดไปตามความยาวของข้อมูลที่ระบุใน 8 ไบต์แรก

◆ เฟรมมาร์คเกอร์ (1 ไบต์)

เฟรมมาร์คเกอร์ใช้สำหรับระบุประเภทเฟรม
'B': เฟรมคำสั่งเลขฐานสอง
'b': เฟรมการตอบสนองเลขฐานสอง
รองรับเฉพาะเฟรมเลขฐานสองเท่านั้น

◆ การควบคุมรายการ (1 ไบต์)



ในขณะรับส่งข้อมูลที่มีขนาดใหญ่ ข้อมูลจะถูกแบ่งออกเป็นหลายเฟรมสำหรับการรับและการส่งข้อมูล บิตควบคุมรายการจะระบุว่าข้อมูลนั้นเป็นเฟรมที่ถูกแบ่งออกมาหรือไม่ ในการระบุว่าเฟรมใดเป็นเฟรมเริ่มต้นหรือเฟรมถัดไป บิตควบคุมจะถูกตั้งค่าเป็น "1" บิตเฟรมสุดท้ายจะเป็น "0"

◆ การควบคุมปลายทาง (1 ไบต์)

เพิ่ม/ไม่เพิ่มบล็อคปลายทาง
ระบุว่าจะเพิ่มบล็อคปลายทางให้กับเฟรมหรือไม่
0: ไม่เพิ่มบล็อคปลายทาง (ไม่ต้องตรวจสอบปลายทาง)
1: เพิ่มบล็อคปลายทาง

ใช้/ไม่ใช้บล็อคปลายทาง
ระบุวิธีการใช้บล็อคปลายทางในการกำหนดโหนดเป้าหมาย
0: เฉพาะโหนดที่มีตำแหน่ง IP
ถูกระบุในบล็อคปลายทางเท่านั้นจึงจะถือว่าเป็นเป้าหมายที่จะทำการประมวลผล
1: เฉพาะโหนดที่มีตำแหน่ง IP
ไม่ถูกระบุในบล็อคปลายทางเท่านั้นจึงจะถือว่าเป็นเป้าหมายที่จะทำการประมวลผล
บิตนี้จะทำงานก็ต่อเมื่อบิตควบคุม “เพิ่ม/ไม่เพิ่มบล็อคปลายทาง” มีค่าเป็น “1” เท่านั้น

โหนดการตอบสนอง
เมื่อต้องการให้มีตอบสนอง บิตนี้จะระบุว่าจะให้โหนดเป้าหมายตอบสนองกลับหรือไม่
0: ให้โหนดทั้งหมดตอบสนองกลับ
1: เฉพาะโหนดที่ถูกระบุที่ส่วนหัวของบล็อคปลายทางเท่านั้นที่จะตอบสนองกลับ
บิตนี้จะทำงานก็ต่อเมื่อบิตควบคุม “เพิ่ม/ไม่เพิ่มบล็อคปลายทาง” และ “ใช้/ไม่ใช้บล็อคปลายทาง” มีค่าเป็น “1” และ “0” ตามลำดับเท่านั้น

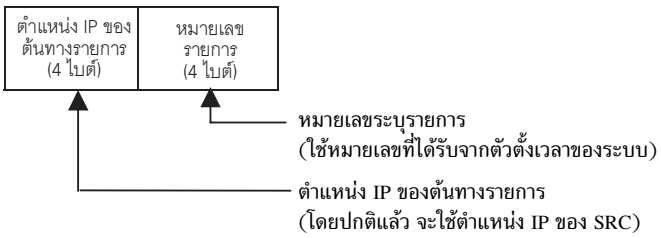
รอการตอบสนอง
บิตนี้จะระบุว่าจะให้โหนดที่ทำตอบสนอง รอเป็นระยะเวลาหนึ่งก่อนที่จะส่งการตอบสนองนั้นหรือไม่
0: ส่งการตอบสนองทันทีโดยไม่ต้องรอเป็นระยะเวลาหนึ่ง
1: รอ

GP จะตอบสนองกลับหลังจากรอเป็นระยะเวลาเท่ากับ “บิตที่มีความสำคัญน้อยที่สุด 7 บิตของตำแหน่ง IP ของ SRC x 1 ms” ฟังก์ชันนี้จะป้องกันไม่ให้มีการตอบสนองจากหลายโหนดพร้อมกัน ในการร้องขอการตอบสนองจากโหนดทั้งหมด ให้ป้อน “09h”

◆ การใช้งานการควบคุมปลายทาง

ในการสื่อสารแบบ 1:1 ตามปกติ ให้ป้อน “00h” ในบิตควบคุมปลายทาง
สำหรับการสื่อสารแบบ “1:n” (มัลติลิงค์) ให้ป้อน “05h” เพื่อร้องขอการตอบสนองจากโหนดเป้าหมายเพียงหนึ่งโหนดจากโหนดต่างๆ ที่ไม่ระบุจำนวน (“n” โหนด) ในการร้องขอการตอบสนองจากโหนดทั้งหมด ให้ป้อน “09h”

◆ **มาร์คเกอร์รายการ**



◆ **การใช้งานของมาร์คเกอร์รายการ**

หลังจากรับเฟรมคำสั่งที่มีมาร์คเกอร์รายการอยู่ด้วยแล้ว GP จะเรียกใช้คำสั่งนั้น (และส่งการตอบสนองกลับหากจำเป็น) กระบวนการนี้จะเป็นเช่นเดียวกันกับการรับเฟรมคำสั่งที่ไม่มีมาร์คเกอร์รายการอยู่ด้วย จากนั้นผลการประมวลผลจะถูกจัดเก็บใน GP
เมื่อ GP ได้รับการร้องขอผลลัพธ์รายการครั้งถัดไปแล้ว GP จะตอบสนองโดยการส่งข้อมูลที่ถูกจัดเก็บไว้ดังกล่าว

GP สามารถจัดเก็บผลลัพธ์รายการได้ถึง 10 รายการ หากมีรายการตั้งแต่ 10 รายการขึ้นไป รายการที่มีอยู่จะถูกลบออก โดยเริ่มจากรายการที่เก่าที่สุด และข้อมูลใหม่จะถูกลงทะเบียน

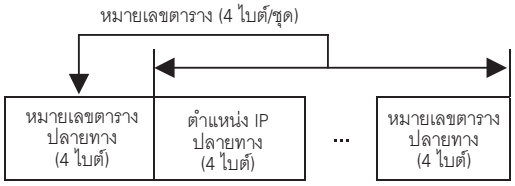
◆ **หมายเลขเฟรม**

เมื่อคำสั่งหรือการตอบสนองหนึ่งถูกแบ่งออกเป็นหลาย ๆ เฟรมแล้ว เฟรมเหล่านั้นจะได้รับการกำหนดหมายเลขเฟรมด้วยเลขลำดับ (เริ่มจาก 0) โดยมีหมายเลขสูงสุดคือ 255

ขนาดใหญ่ที่สุดของเฟรมที่ถูกแบ่งออกมาคือ 1 กิโลไบต์

◆ **บล็อคปลายทาง**

บล็อคปลายทางจะถูกเพิ่มเข้ามาหากบิตควบคุม “เพิ่ม/ไม่เพิ่มบล็อคปลายทาง” มีค่าเป็น “1” บล็อคปลายทางจะไม่ถูกเพิ่มเข้ามาหากบิตควบคุมนี้มีค่าเป็น “0”



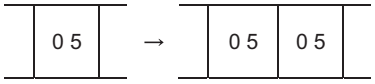
2.2 SIO 1: หมายเหตุเกี่ยวกับการสื่อสารแบบ N Binary

ในการสื่อสารแบบ 1:N binary ของโหมดขยาย SIO จะมีกระบวนการคู่เกิดขึ้น

2.2.1 การถ่ายโอนข้อมูลจากโฮสต์ไปที่ GP

■ ENQ

ในการรับส่งข้อมูลจากโฮสต์ หากข้อมูลสำหรับ “ช่วงการตรวจสอบผลรวม” หรือ “การตรวจสอบผลรวม” มีค่า “05h” ซึ่งเหมือนกับรหัส ENQ แล้ว ให้เพิ่ม “05h” เข้าไปก่อนที่จะส่งข้อมูล

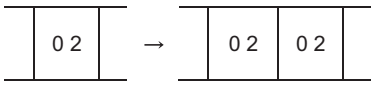


โปรดทราบว่า “05h” ที่เพิ่มให้นี้ไม่มีอยู่ในหมายเลขข้อมูล หากพื้นที่เก็บข้อมูลคำสั่งมี “หมายเลขข้อมูล”

2.2.2 การถ่ายโอนข้อมูลจากโฮสต์ไปที่ GP

■ STX

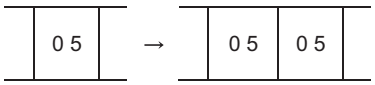
ในการตอบสนองที่ส่งมาจาก GP series หากข้อมูลสำหรับ “ช่วงการตรวจสอบผลรวม” หรือ “การตรวจสอบผลรวม” มีค่า “02h” ซึ่งเหมือนกับรหัส STX แล้ว ให้เพิ่ม “02h” เข้าไปก่อนที่จะทำการตอบสนอง



โปรดทราบว่า “02h” ที่เพิ่มให้นี้ไม่มีอยู่ในหมายเลขข้อมูล หากพื้นที่เก็บข้อมูลคำสั่งมี “หมายเลขข้อมูล”

■ ENQ

สำหรับการเชื่อมต่อแบบ 1:N ชนิด 2 สาย หากข้อมูลสำหรับ “ช่วงการตรวจสอบผลรวม” หรือ “การตรวจสอบผลรวม” มีค่า “05h” ซึ่งเหมือนกับรหัส ENQ ในการตอบสนองจาก GP series แล้ว ให้เพิ่ม “05h” เข้าไปก่อนที่จะส่งข้อมูล



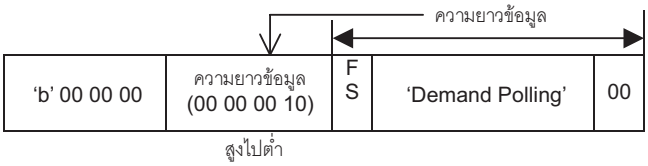
2.3 การโพลดีมานด์

ในการเชื่อมต่อแบบ TCP หาก GP ไม่ได้รับการร้องขอเป็นระยะ ๆ จากโฮสต์แล้ว เครื่อง GP จะตรวจสอบว่ามีโฮสต์อยู่หรือไม่ด้วยการโพล์ดีมานด์
เมื่อโฮสต์ได้รับการร้องขอนี้แล้ว ให้ตรวจสอบว่าได้ส่งคำขอโพลดีมานด์เดียวกันนี้ให้กับ GP แล้ว หลังจาก GP ได้รับการร้องขอนี้แล้ว จะเป็นการยืนยันว่ามีโฮสต์อยู่
หากไม่ได้รับการตอบสนองจากโฮสต์ GP จะปิดการเชื่อมต่อ
หากคุณต้องการนำ API ต่าง ๆ ของ Digital Electronics Corporation มาใช้กับโฮสต์แล้ว API ดังกล่าวจะประมวลผลการตอบสนองของการร้องขอโพลดีมานด์โดยอัตโนมัติ

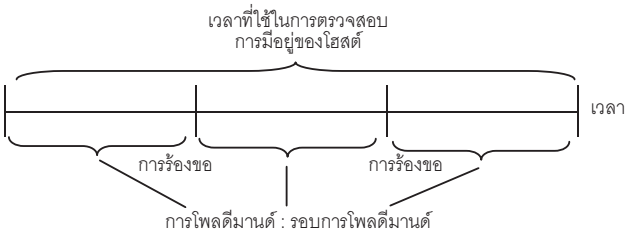
2.3.1 การโพลดีมานด์ (ดีมานด์ FS)

รายละเอียดข้อมูลของการร้องขอโพลดีมานด์ที่ถูกส่งจาก GP ไปยังโฮสต์มีดังต่อไปนี้:

โฮสต์: ไม่มีข้อมูลใด ๆ
GP: ข้อมูลการตอบสนอง



ชื่อข้อมูล
- ข้อมูล: "Demand Polling"



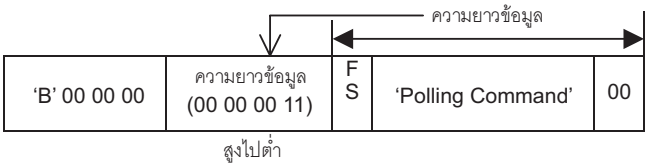
การโพลดีมานด์: เมื่อครบกำหนดเวลาตรวจสอบการมีอยู่ของโฮสต์แล้ว การร้องขอโพลดีมานด์จะถูกส่งออกไป การร้องขอนี้ต้องการให้โฮสต์ส่งคำสั่งการโพลของตนออกมา การร้องขอชนิดนี้ทำให้ GP ไม่ต้องรอให้มีการร้องขอการโพล

โฮสต์						
GP	'b' 00 00 00	00 00 00 10	F S	'Demand Polling'	00

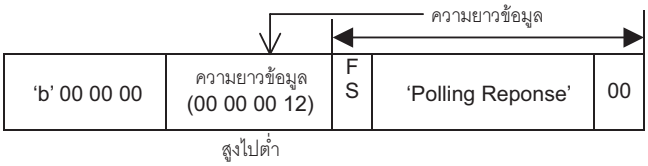
2.3.2 คำสั่งการโพล (การโพล FS)

รายละเอียดข้อมูลของคำสั่งการโพลที่ถูกส่งจากโฮสต์ไปยัง GP มีดังต่อไปนี้:

โฮสต์: ข้อมูลคำสั่ง



GP: ข้อมูลการตอบสนอง



ชื่อข้อมูล

- **ข้อมูล: "Polling Command"**

ตัวอย่าง:

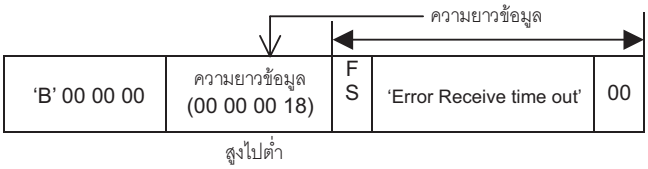
โฮสต์	'B' 00 00 00	00000011	F S	'Polling Command'	00					
GP						'b' 00 00 00	00000012	F S	'Polling Response'	00

2.3.3 การตรวจหาข้อผิดพลาด (ข้อผิดพลาด FS)

เมื่อเกิดข้อผิดพลาดของโปรโตคอล คำสั่งนี้ทำให้ GP หรือโฮสต์สามารถส่งการแจ้งข้อผิดพลาดของยูนิต/อุปกรณ์อื่นได้
เฟรมนี้ไม่ต้องมีการตอบสนอง

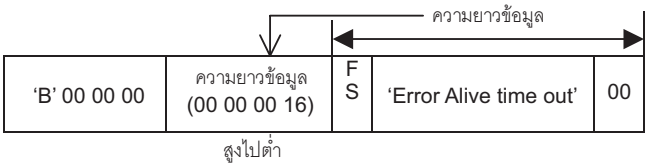
โฮสต์: ข้อมูลคำสั่ง

- **เฟรมข้อผิดพลาดไทม์เอาต์ของ** Inter-character



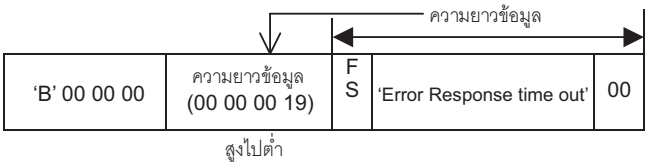
ชื่อข้อมูล

- **ข้อมูล:** "Error Receive time out"
- **GP แสดงเฟรมข้อผิดพลาดไทม์เอาต์การตรวจสอบ**



ชื่อข้อมูล

- **ข้อมูล:** "Error Alive time out"
- **เฟรมข้อผิดพลาดไทม์เอาต์ของ** Inter-protocol

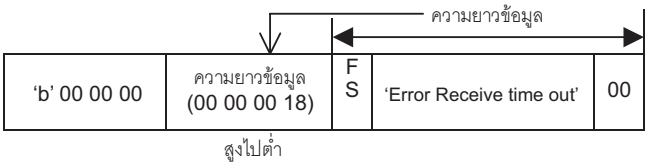


ชื่อข้อมูล

- **ข้อมูล:** "Error Response time out"

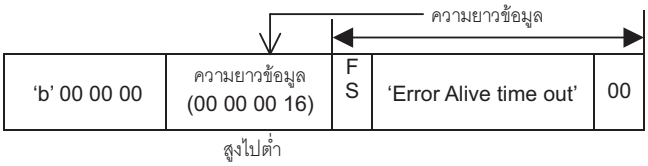
GP: ข้อมูลการตอบสนอง

- **เฟรมข้อผิดพลาดไทม์เอาต์ของ** Inter-character



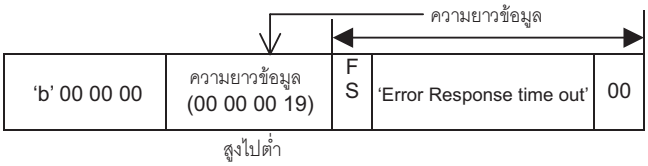
ชื่อข้อมูล

- **ข้อมูล:** “Error Receive time out”
- **GP แสดงเฟรมข้อผิดพลาดไทม์เอาต์การตรวจสอบ**



ชื่อข้อมูล

- **ข้อมูล:** “Error Alive time out”
- **เฟรมข้อผิดพลาดไทม์เอาต์ของ** Inter-protocol



ชื่อข้อมูล

- **ข้อมูล:** “Error Receive time out”

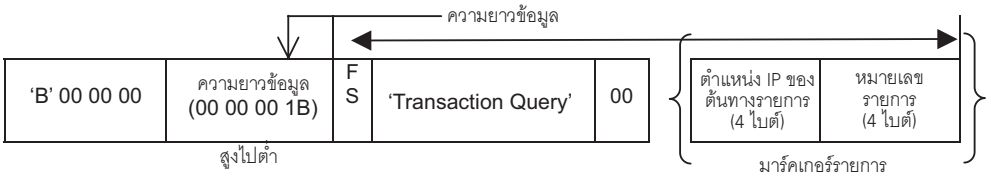
2.4 คำสั่งร้องขอผลลัพธ์รายการ

คำสั่งนี้สามารถใช้ได้เฉพาะเมื่อมีการใช้ Ethernet

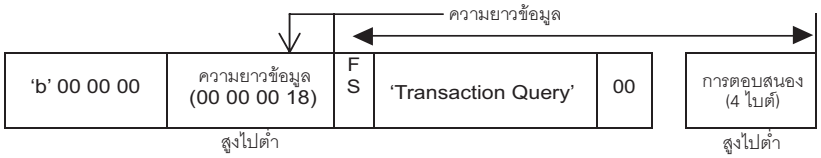
2.4.1 การร้องขอผลลัพธ์รายการ

เนื้อหาของการร้องขอผลลัพธ์รายการที่ส่งจากอุปกรณ์โฮสต์ไปที่ GP มีดังต่อไปนี้:

โฮสต์: ข้อมูลคำสั่ง



GP: ข้อมูลการตอบสนอง



ชื่อข้อมูล

- ข้อมูล: 'Transaction Query'

ผลลัพธ์ที่เรียกใช้

- ข้อมูล: 'Transaction Result'

ค่า	ความหมาย
0x00000000	การประมวลผลสิ้นสุดอย่างถูกต้อง
0x00000001	ข้อผิดพลาด
0x00000002	ไม่ได้จัดเก็บมาร์คเกอร์รายการที่กำหนดไว้ใน GP

2.5 รูปแบบคำสั่ง

2.5.1 รูปแบบการอ่านข้อมูล

■ **โหมดการแปลง** SIO

◆ **บล็อคข้อมูลคำสั่ง (จากโฮสต์)**

ESC	R	ตำแหน่ง (4 ไบต์)	จำนวนของข้อมูล (4 ไบต์)	CR

ช่วงการตั้งค่า
ตำแหน่ง: 0000H ถึง 270FH (0 ถึง 9999)
จำนวนแพ็กเก็ตข้อมูล: 0001H ถึง 0100H (1 ถึง 256)

โปรดตรวจสอบว่าได้ป้อนข้อมูลทั้งหมดในรูปแบบรหัส ASCII

◆ **บล็อคข้อมูลการตอบสนองของ** GP **(จาก** GP**)**

- **เมื่อไม่มีข้อผิดพลาด**

ESC	A	แพ็กเก็ตข้อมูล 1 (4 ไบต์)	...	แพ็กเก็ตข้อมูล n (4 ไบต์)	CR

<ช่วงการตั้งค่า>
ข้อมูล: 0000H ถึง FFFFH

- **เมื่อเกิดข้อผิดพลาด**

การตอบสนอง NAK

■ **โหมดขยาย** SIO, ASCII

◆ **บล็อคข้อมูลคำสั่ง (จากโฮสต์)**

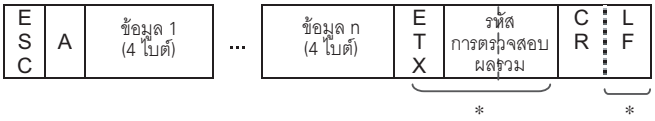
ESC	R	ตำแหน่ง (4 ไบต์)	จำนวนของข้อมูล (4 ไบต์)	รหัสการตรวจสอบผลรวม	CR	LF
				*		*

อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
<ช่วงการตั้งค่า>
ตำแหน่ง: 0000H ถึง 270FH (0 ถึง 9999)
จำนวนแพ็กเก็ตข้อมูล: 0001H ถึง 0100H (1 ถึง 256)

โปรดตรวจสอบว่าได้ป้อนข้อมูลทั้งหมดในรูปแบบรหัส ASCII

◆ **บล็อคข้อมูลการตอบสนองของ GP (จาก GP)**

- **เมื่อไม่มีข้อผิดพลาด**



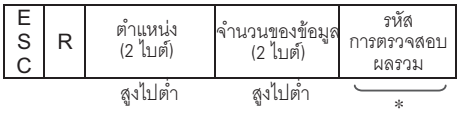
อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
<ช่วงการตั้งค่า>
ข้อมูล: 0000H ถึง FFFFH

- **เมื่อเกิดข้อผิดพลาด**

การตอบสนอง NAK

■ โหมดขยาย SIO, เลขฐานสอง

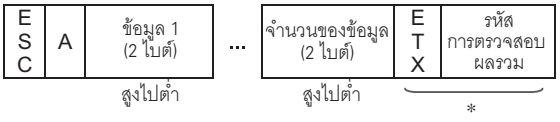
◆ **บล็อคข้อมูลคำสั่ง (จากโฮสต์)**



อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
<ช่วงการตั้งค่า>
ตำแหน่ง: 0000H ถึง 270FH (0 ถึง 9999)
จำนวนแพ็กเก็ตข้อมูล: 0001H ถึง 0200H (1 ถึง 512)

◆ **บล็อคข้อมูลการตอบสนองของ GP (จาก GP)**

- **เมื่อไม่มีข้อผิดพลาด**



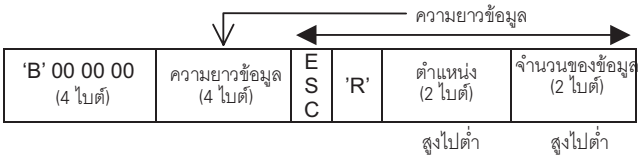
อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
<ช่วงการตั้งค่า>
ข้อมูล: 0000H ถึง FFFFH

- **เมื่อเกิดข้อผิดพลาด**

การตอบสนอง NAK

■ LAN

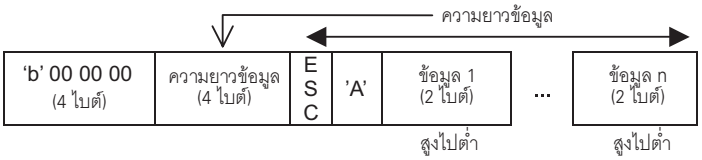
◆ **บล็อคข้อมูลคำสั่ง (จากโฮสต์)**



<ช่วงการตั้งค่า>
ตำแหน่ง: 0000H ถึง 270FH (0 ถึง 9999)
จำนวนแพ็กเก็ตข้อมูล: 0001H ถึง 0200H (1 ถึง 512)

◆ **บล็อคข้อมูลการตอบสนองของ GP (จาก GP)**

- **เมื่อไม่มีข้อผิดพลาด**



<ช่วงการตั้งค่า>
ข้อมูล: 0000H ถึง FFFFH

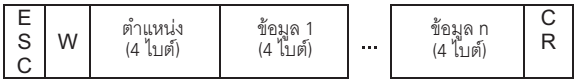
- **เมื่อเกิดข้อผิดพลาด**

การตอบสนอง NAK

2.5.2 รูปแบบการเขียนข้อมูล

■ โหมดการแปลง SIO

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)

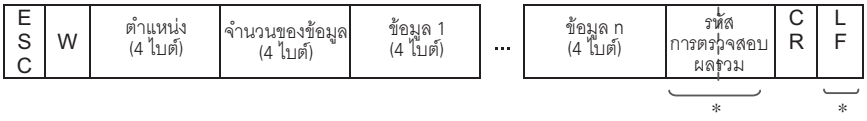


<ช่วงการตั้งค่า>
ตำแหน่ง: 0000H ถึง 270FH (0 ถึง 9999)
ข้อมูล: 0000H ถึง FFFFH

โปรดตรวจสอบว่าได้ป้อนข้อมูลทั้งหมดในรูปแบบรหัส ASCII
ในโหมดการแปลง จะไม่มีคำสั่งการตอบสนองจาก GP
ในโหมดการแปลง จะไม่มีการจำกัดจำนวนแพ็กเก็ตข้อมูลการเขียน

■ โหมดขยาย SIO, ASCII

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)



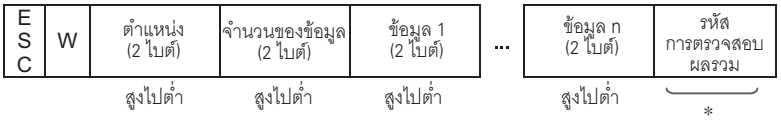
อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
<ช่วงการตั้งค่า>
ตำแหน่ง: 0000H ถึง 270FH (0 ถึง 9999)
จำนวนแพ็กเก็ตข้อมูล: 0001H ถึง 0100H (1 ถึง 256)
ข้อมูล: 0000H ถึง FFFFH

โปรดตรวจสอบว่าได้ป้อนข้อมูลทั้งหมดในรูปแบบรหัส ASCII

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)
การตอบสนอง ACK หรือ NAK

■ โหมดขยาย SIO, เลขฐานสอง

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)



อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า
<ช่วงการตั้งค่า>
ตำแหน่ง: 0000H ถึง 270FH (0 ถึง 9999)
จำนวนแพ็กเก็ตข้อมูล: 0001H ถึง 0200H (1 ถึง 512)
ข้อมูล: 0000H ถึง FFFFH

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)

การตอบสนอง ACK หรือ NAK

■ LAN

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)

<ช่วงการตั้งค่า>
ตำแหน่ง: 0000H ถึง 270FH (0 ถึง 9999)
จำนวนแพ็กเก็ตข้อมูล: 0001H ถึง 0200H (1 ถึง 512)
ข้อมูล: 0000H ถึง FFFFH

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)

การตอบสนอง ACK หรือ NAK

2.5.3 การร้องขอส่งออกสัญญาณอินเตอร์รัพต์

ในที่นี้จะอธิบายคำสั่งต่าง ๆ ที่ GP ใช้ในโหมดขยาย เมื่อสื่อสารแบบ [1:n ASCII], [1:n Binary] หรือ 2 wire เพื่อส่งออกรหัสสัญญาณอินเตอร์รัพต์ผ่านทางการเขียนค่าสัมบูรณ์หรือค่าอื่น ๆ ของพื้นที่ระบบของ GP โดยส่งจาก GP ไปที่โฮสต์
เมื่อทำการเชื่อมต่อแบบ 2 wire ให้ทำการตั้งค่าดังต่อไปนี้ แม้จะเป็นการเชื่อมต่อแบบ 1:1 ก็ตาม

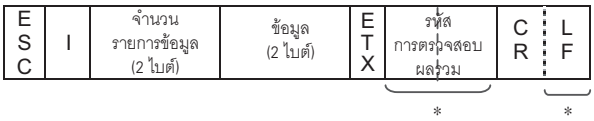
■ SIO, โหมด ASCII

◆ **บล็อคข้อมูลคำสั่ง (จากโฮสต์)**

อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

◆ **บล็อคข้อมูลการตอบสนองของ GP (จาก GP)**

- **เมื่อไม่มีข้อผิดพลาด**



- **เมื่อเกิดข้อผิดพลาด**
  การตอบสนอง NAK

<ช่วงการตั้งค่า>
จำนวนแพ็กเก็ตข้อมูล
เมื่อคำสั่งร้องขอถูกส่งออกจากโฮสต์ ค่านี้จะเป็นตัวกำหนดจำนวนของรายการข้อมูลของการส่งออกสัญญาณอินเตอร์รัพต์ที่ออกก่อนหน้า
เมื่อได้รับข้อมูลของการส่งออกสัญญาณอินเตอร์รัพต์ที่ออกก่อนหน้าทั้งหมดแล้ว จะต้องส่งความถี่ (จำนวน) ข้อมูลนี้ออกไป
ข้อมูล
ข้อมูล (00H ถึง FEH) จะถูกแปลงเป็นรหัส ASCII ชนิด 2 หลักก่อนที่จะถูกส่งออกไป
ฟิลด์นี้จะถูกป้อนด้วย “00” หากไม่มีข้อมูลที่จะส่งออก

■ SIO, โหมดเลขฐานสอง

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)

ESC	I	รหัสการตรวจสอบผลรวม	
		*	

อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)

- เมื่อไม่มีข้อผิดพลาด

ESC	I	จำนวนรายการข้อมูล (1 ไบต์)	ข้อมูล (1 ไบต์)	ETX	รหัสการตรวจสอบผลรวม
					*

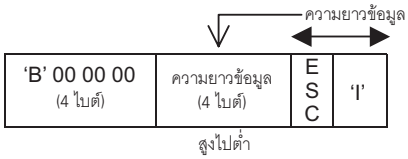
อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

- เมื่อเกิดข้อผิดพลาด
  การตอบสนอง NAK

<ช่วงการตั้งค่า>
จำนวนแพ็กเก็ตข้อมูล
เมื่อคำสั่งร้องขอถูกส่งออกจากโฮสต์ ค่านี้จะเป็นตัวกำหนดจำนวนของรายการข้อมูลของการส่งออกสัญญาณอินเตอร์รัพต์ที่ออกก่อนหน้า
เมื่อได้รับข้อมูลของการส่งออกสัญญาณอินเตอร์รัพต์ที่ออกก่อนหน้าทั้งหมดแล้ว จะต้องส่งความถี่ (จำนวน) ข้อมูลนี้ออกไป
ข้อมูล
ค่าข้อมูล (00H ถึง FEH) จะถูกส่งออกไป
ฟิลด์นี้จะถูกป้อนด้วย “00” หากไม่มีข้อมูลที่จะส่งออก

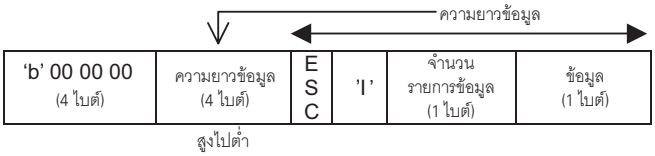
■ LAN

◆ **บล็อคข้อมูลคำสั่ง (จากโฮสต์)**



◆ **บล็อคข้อมูลการตอบสนองของ GP (จาก GP)**

- **เมื่อไม่มีข้อผิดพลาด**



- **เมื่อเกิดข้อผิดพลาด**
  การตอบสนอง NAK

<ช่วงการตั้งค่า>

จำนวนแพ็กเก็ตข้อมูล

เมื่อคำสั่งสอบถามถูกส่งออกจากโฮสต์ ค่านี้จะเป็นตัวกำหนดจำนวนของรายการข้อมูลของการส่งออกสัญญาณอินเตอร์รัพต์ที่ออกก่อนหน้า

ข้อมูล

ค่าข้อมูล (00H ถึง FEH) จะถูกส่งออกไป
ฟิลด์นี้จะถูกป้อนด้วย "00" หากไม่มีข้อมูลที่จะส่งออก

2.5.4 การปรับความสว่างและคอนทราสต์

ด้านล่างนี้คือรูปแบบของบล็อคข้อมูลคำสั่งที่มีคำสั่ง ESC # (คำสั่งปรับความสว่างและคอนทราสต์)
โปรดทราบว่าใน GP บางชนิด คุณไม่สามารถปรับความสว่างหรือคอนทราสต์ได้

■ โหมดการแปลง SIO

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)

ESC	#	ลักษณะเฉพาะ (4 ไบต์)	การตั้งค่า (4 ไบต์)	CR

<ช่วงการตั้งค่า>
ลักษณะเฉพาะ: 0000H ถึง 0001H (0: คอนทราสต์, 1: ความสว่าง)
การตั้งค่า: โปรดดูที่ “■ ตารางความสว่าง/คอนทราสต์” (หน้า 45)

โปรดตรวจสอบว่าได้ป้อนข้อมูลทั้งหมดในรูปแบบรหัส ASCII

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)

ไม่มีข้อมูลการตอบสนอง

■ โหมดขยาย SIO, ASCII

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)

ESC	#	ลักษณะเฉพาะ (4 ไบต์)	การตั้งค่า (4 ไบต์)	รหัสการตรวจสอบผลรวม	CR	LF
				*		*

อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

<ช่วงการตั้งค่า>
ลักษณะเฉพาะ: 0000H ถึง 0001H (0: คอนทราสต์, 1: ความสว่าง)
การตั้งค่า: โปรดดูที่ “■ ตารางความสว่าง/คอนทราสต์” (หน้า 45)

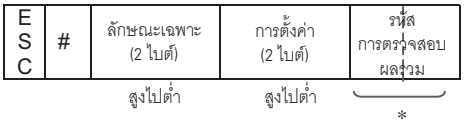
โปรดตรวจสอบว่าได้ป้อนข้อมูลทั้งหมดในรูปแบบรหัส ASCII

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)

การตอบสนอง ACK หรือ NAK

■ โหมดขยาย SIO, เลขฐานสอง

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)



<ช่วงการตั้งค่า>
ลักษณะเฉพาะ: 0000H ถึง 0001H (0: คอนทราสต์, 1: ความสว่าง)
การตั้งค่า: โปรดดูที่ “■ ตารางความสว่าง/คอนทราสต์” (หน้า 45)

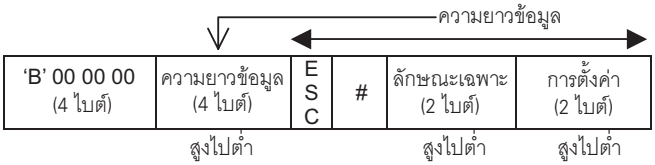
โปรดตรวจสอบว่าได้ป้อนข้อมูลทั้งหมดในรูปแบบรหัส ASCII

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)

การตอบสนอง ACK หรือ NAK

■ LAN

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)



<ช่วงการตั้งค่า>
ลักษณะเฉพาะ: 0000H ถึง 0001H (0: คอนทราสต์, 1: ความสว่าง)
การตั้งค่า: โปรดดูที่ “■ ตารางความสว่าง/คอนทราสต์” (หน้า 45)

โปรดตรวจสอบว่าได้ป้อนข้อมูลทั้งหมดในรูปแบบรหัส ASCII

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)

การตอบสนอง ACK หรือ NAK

2.5.5 ค่าปัจจุบันของความสว่างและคอนทราสต์

ด้านล่างนี้คือรูปแบบของบล็อคข้อมูลคำสั่งเพื่อรับค่าปัจจุบันของความสว่างและคอนทราสต์ด้วยคำสั่งดังกล่าว โปรดทราบว่าใน GP บางชนิด คุณไม่สามารถปรับระดับความสว่างหรือคอนทราสต์ได้

■ **โหมดการแปลง** SIO

◆ **บล็อคข้อมูลคำสั่ง (จากโฮสต์)**

ESC	$	CR

◆ **บล็อคข้อมูลการตอบสนองของ** GP **(จาก** GP**)**

ESC	D	ระดับคอนทราสต์ (4 ไบต์)	ระดับความสว่าง (4 ไบต์)	CR

■ **โหมดขยาย** SIO, ASCII

◆ **บล็อคข้อมูลคำสั่ง (จากโฮสต์)**

ESC	$	รหัสการตรวจสอบผลรวม	CR	LF
		*		*

อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

◆ **บล็อคข้อมูลการตอบสนองของ** GP **(จาก** GP**)**

ESC	D	ระดับคอนทราสต์ (4 ไบต์)	ระดับความสว่าง (4 ไบต์)	รหัสการตรวจสอบผลรวม	CR	LF
				*		*

อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

■ โหมดขยาย SIO, เลขฐานสอง

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)

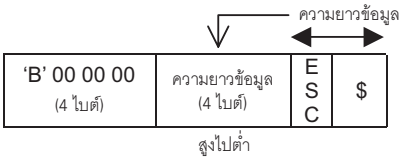
อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)

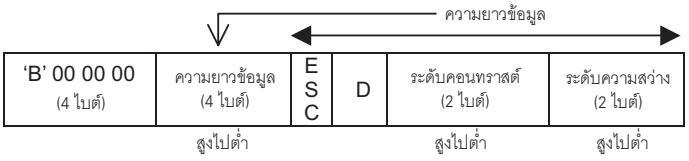
อาจไม่ต้องเพิ่มพื้นที่ที่มีเครื่องหมายดอกจัน (*) ก็ได้ ทั้งนี้ขึ้นอยู่กับการตั้งค่า

■ LAN

◆ บล็อคข้อมูลคำสั่ง (จากโฮสต์)



◆ บล็อคข้อมูลการตอบสนองของ GP (จาก GP)



■ ตารางความสว่าง/คอนทราสต์

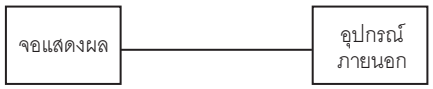
GP	ความสว่าง ช่วงการตั้งค่า	คอนทราสต์ ช่วงการตั้งค่า
AGP-3302B	0(สว่าง) ถึง 7(มืด)	0(สว่าง) ถึง 7(มืด)
AGP-3301L	0(สว่าง) ถึง 7(มืด)	0(สว่าง) ถึง 7(มืด)
AGP-3301S	0(สว่าง) ถึง 7(มืด)	0(สว่าง) ถึง 7(มืด)
AGP-3300L	0(สว่าง) ถึง 7(มืด)	0(สว่าง) ถึง 7(มืด)
AGP-3300S	0(สว่าง) ถึง 7(มืด)	0(สว่าง) ถึง 7(มืด)
AGP-3300T	0(สว่าง) ถึง 7(มืด)	-
AGP-3400S	0(สว่าง) ถึง 7(มืด)	0(สว่าง) ถึง 7(มืด)
AGP-3400T	0(สว่าง) ถึง 7(มืด)	-
AGP-3500L	0(สว่าง) ถึง 7(มืด)	0(สว่าง) ถึง 7(มืด)
AGP-3500S	0(สว่าง) ถึง 7(มืด)	0(สว่าง) ถึง 7(มืด)
AGP-3500T	0(สว่าง) ถึง 7(มืด)	-
AGP-3600T	0(สว่าง) ถึง 7(มืด)	-
AGP-3450T	0(สว่าง) ถึง 7(มืด)	-
AGP-3550T	0(สว่าง) ถึง 7(มืด)	-
AGP-3650T	0(สว่าง) ถึง 7(มืด)	-
AGP-3750T	0(สว่าง) ถึง 7(มืด)	-

3 รายละเอียดของระบบ

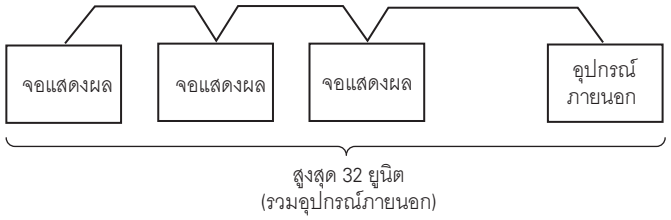
เมื่อใช้การเชื่อมต่อผ่านหน่วยความจำ รายละเอียดในการเชื่อมต่อจะเป็นดังนี้

■ อนุกรม

- การเชื่อมต่อแบบ 1:1

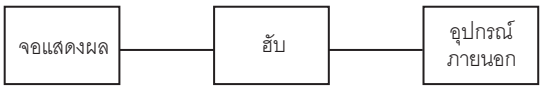


การเชื่อมต่อแบบ 1:n

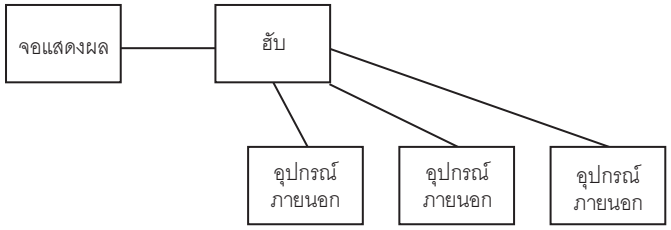


■ Ethernet

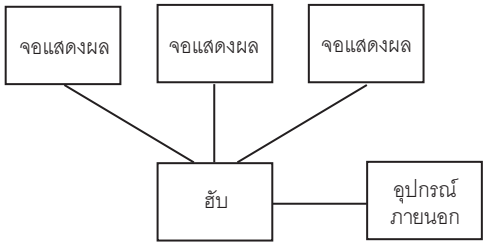
- การเชื่อมต่อแบบ 1:1



- การเชื่อมต่อแบบ 1:n

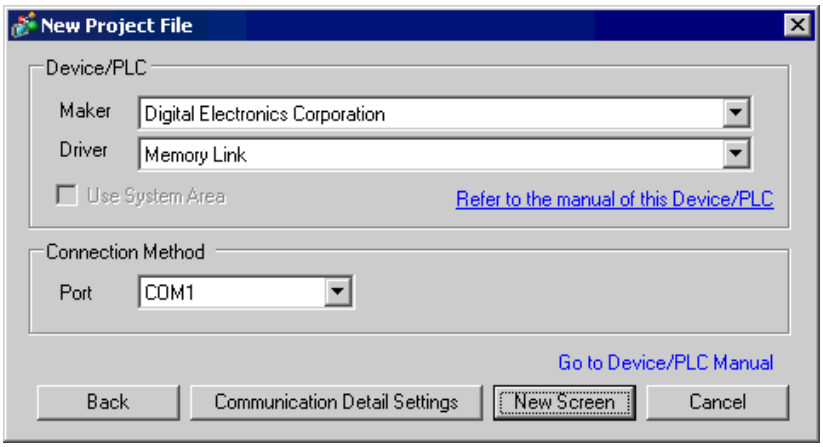


- การเชื่อมต่อแบบ n:1 (มีเฉพาะเมื่อเลือก “Ethernet (UDP)”)



4 การเลือกไดรเวอร์

เลือกอุปกรณ์ภายนอกที่จะนำมาเชื่อมต่อกับจอแสดงผล



รายการตั้งค่า	คำอธิบายการตั้งค่า
Maker	เลือกผู้ผลิตอุปกรณ์ภายนอกที่จะใช้เชื่อมต่อ เลือก “Digital Electronics Corporation”
Driver	เลือกรหัส (รุ่น) ของอุปกรณ์ภายนอกที่จะเชื่อมต่อและวิธีเชื่อมต่อ เลือก “Memory Link” คุณสามารถตรวจสอบรายละเอียดการเชื่อมต่อใน “การเชื่อมต่อผ่านหน่วยความจำ” ได้จาก รายละเอียดของระบบ ☞ “รายละเอียดของระบบ” (หน้า 46)
Port	เลือกพอร์ตการแสดงผลที่จะเชื่อมต่อกับอุปกรณ์ภายนอก

5 ตัวอย่างการตั้งค่าการสื่อสาร

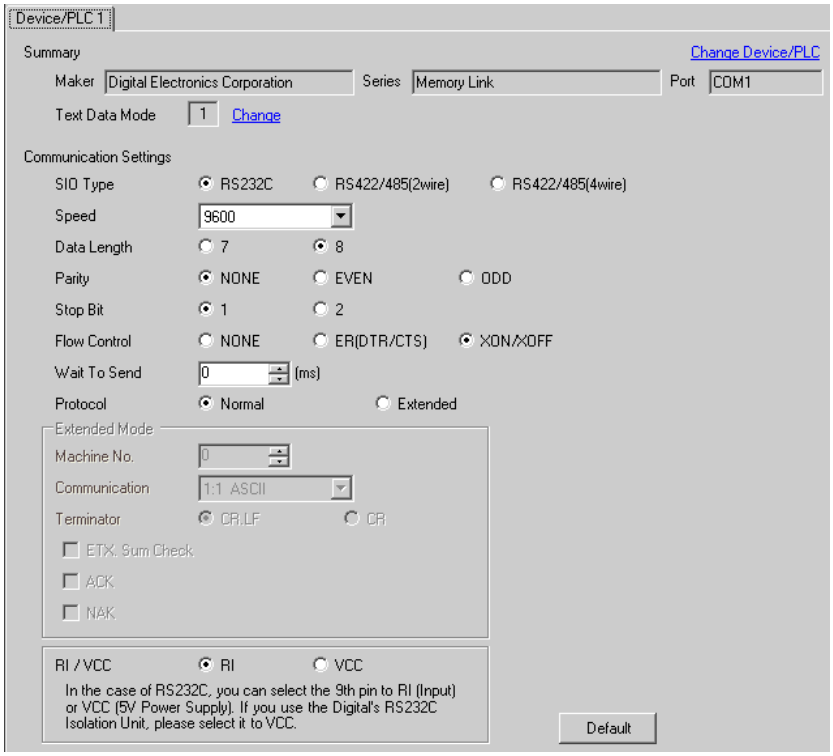
ตัวอย่างการตั้งค่าการสื่อสารของจอแสดงผลและอุปกรณ์ภายนอกตามที่ Pro-face แนะนำ

5.1 ตัวอย่างการตั้งค่าที่ 1

■ การตั้งค่าของ GP-Pro EX (การเชื่อมต่อแบบ RS232C: โหมดการแปลง)

◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



■ การตั้งค่าอุปกรณ์ภายนอก

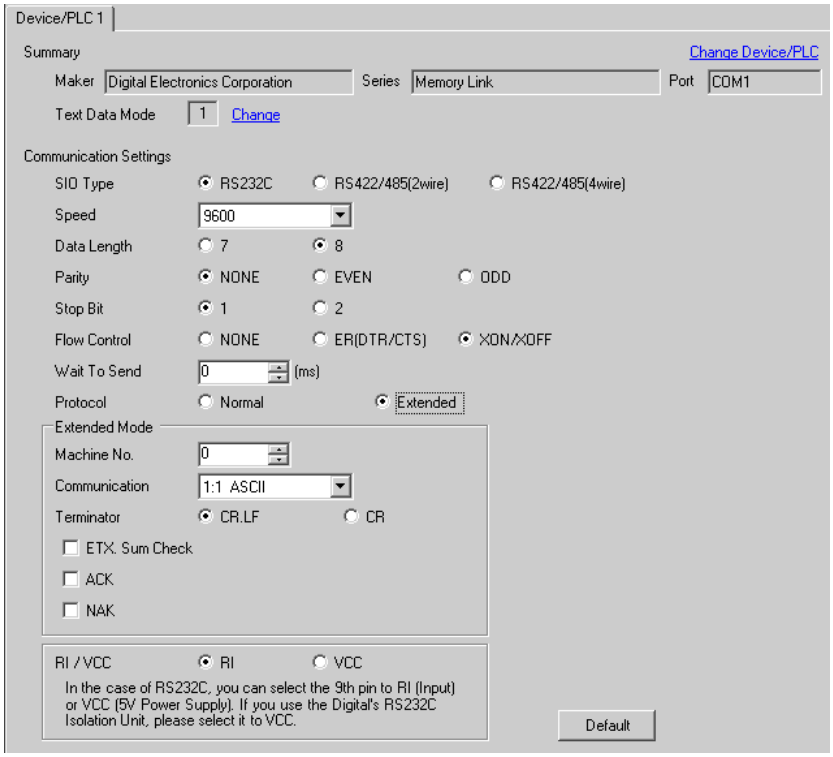
ขึ้นอยู่กับข้อมูลจำเพาะของอุปกรณ์ภายนอก

5.2 ตัวอย่างการตั้งค่าที่ 2

■ การตั้งค่าของ GP-Pro EX (การเชื่อมต่อแบบ RS232C: โหมดขยาย)

◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



■ การตั้งค่าอุปกรณ์ภายนอก

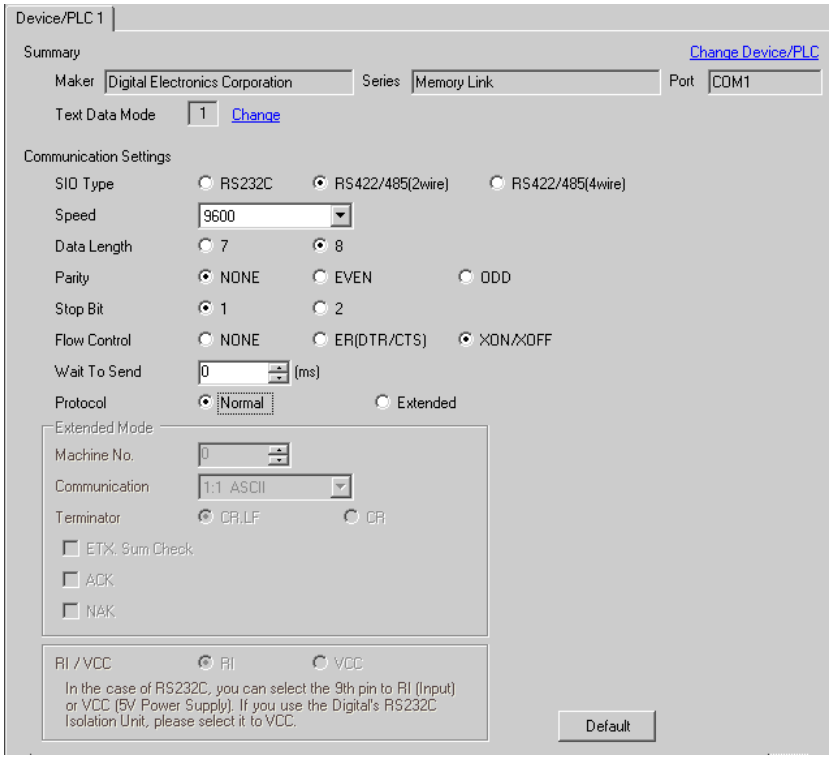
ขึ้นอยู่กับข้อมูลจำเพาะของอุปกรณ์ภายนอก

5.3 ตัวอย่างการตั้งค่าที่ 3

■ การตั้งค่าของ GP-Pro EX (การเชื่อมต่อแบบ RS422/485 (2wire): โหมดการแปลง)

◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



■ การตั้งค่าอุปกรณ์ภายนอก

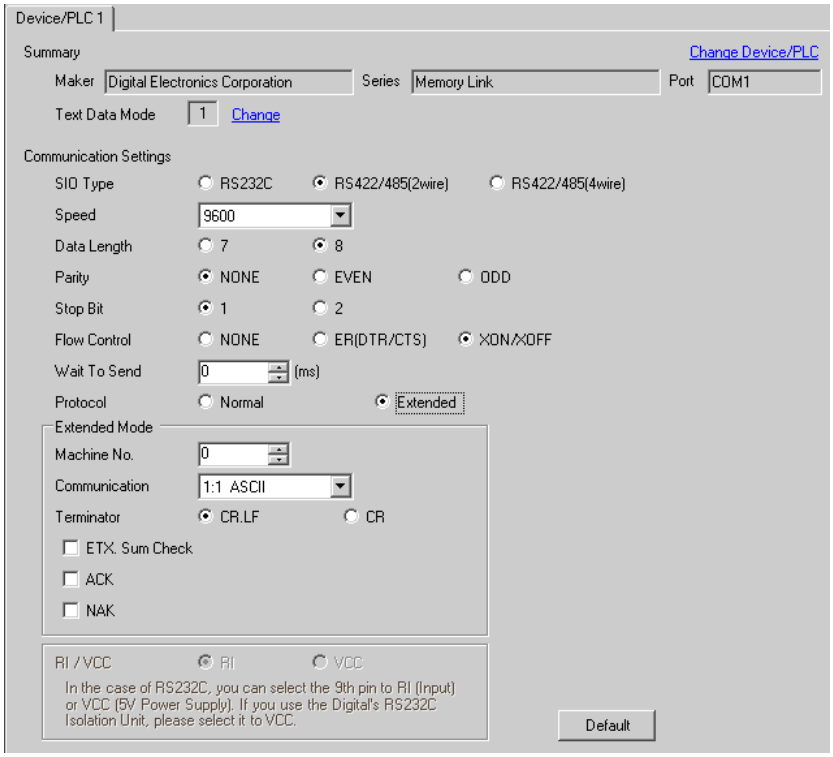
ขึ้นอยู่กับข้อมูลจำเพาะของอุปกรณ์ภายนอก

5.4 ตัวอย่างการตั้งค่าที่ 4

■ การตั้งค่าของ GP-Pro EX (การเชื่อมต่อแบบ RS422/485 (2wire): โหมดขยาย)

◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



■ การตั้งค่าอุปกรณ์ภายนอก

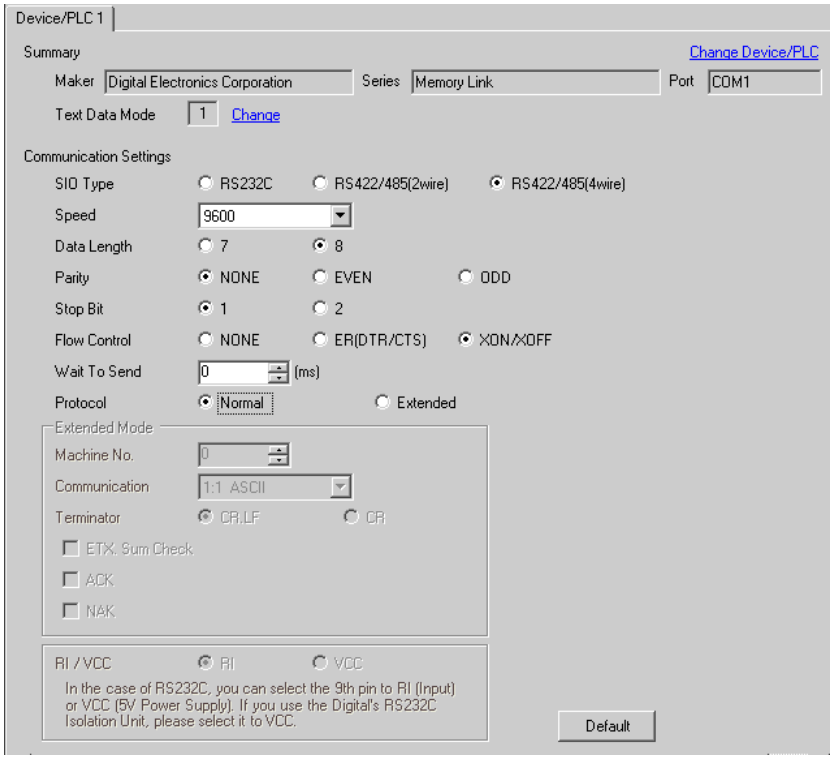
ขึ้นอยู่กับข้อมูลจำเพาะของอุปกรณ์ภายนอก

5.5 ตัวอย่างการตั้งค่าที่ 5

■ การตั้งค่าของ GP-Pro EX (การเชื่อมต่อแบบ RS422/485 (4wire): โหมดการแปลง)

◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



■ การตั้งค่าอุปกรณ์ภายนอก

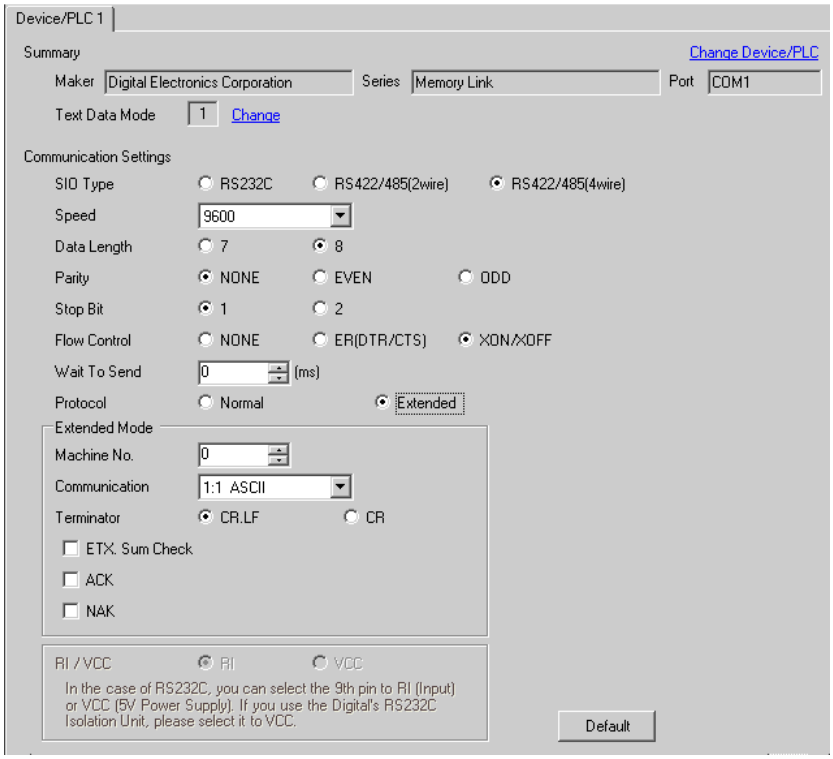
ขึ้นอยู่กับข้อมูลจำเพาะของอุปกรณ์ภายนอก

5.6 ตัวอย่างการตั้งค่าที่ 6

■ การตั้งค่าของ GP-Pro EX (การเชื่อมต่อแบบ RS422/485 (4wire): โหมดขยาย)

◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



■ การตั้งค่าอุปกรณ์ภายนอก

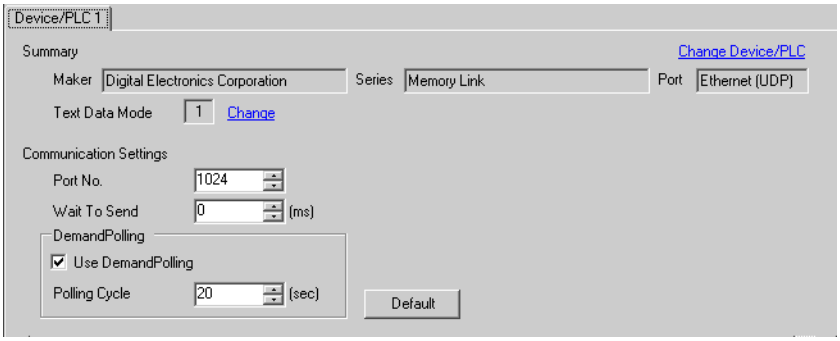
ขึ้นอยู่กับข้อมูลจำเพาะของอุปกรณ์ภายนอก

5.7 ตัวอย่างการตั้งค่าที่ 7

■ การตั้งค่าของ GP-Pro EX (การเชื่อมต่อแบบ Ethernet (UDP))

◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



■ การตั้งค่าอุปกรณ์ภายนอก

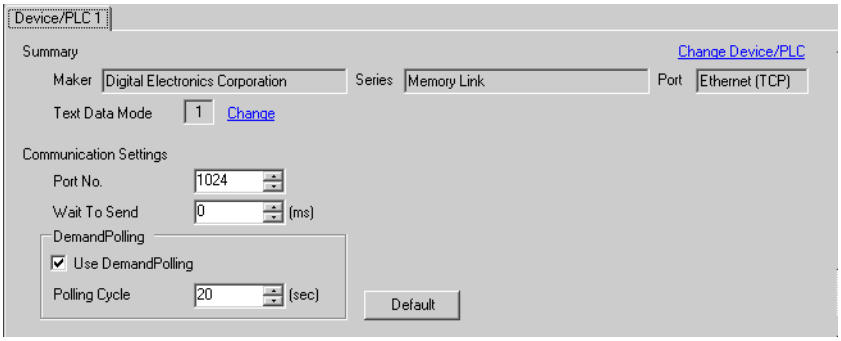
ขึ้นอยู่กับข้อมูลจำเพาะของอุปกรณ์ภายนอก

5.8 ตัวอย่างการตั้งค่าที่ 8

■ การตั้งค่าของ GP-Pro EX (การเชื่อมต่อแบบ Ethernet (TCP))

◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



■ การตั้งค่าอุปกรณ์ภายนอก

ขึ้นอยู่กับข้อมูลจำเพาะของอุปกรณ์ภายนอก

6 รายการตั้งค่า

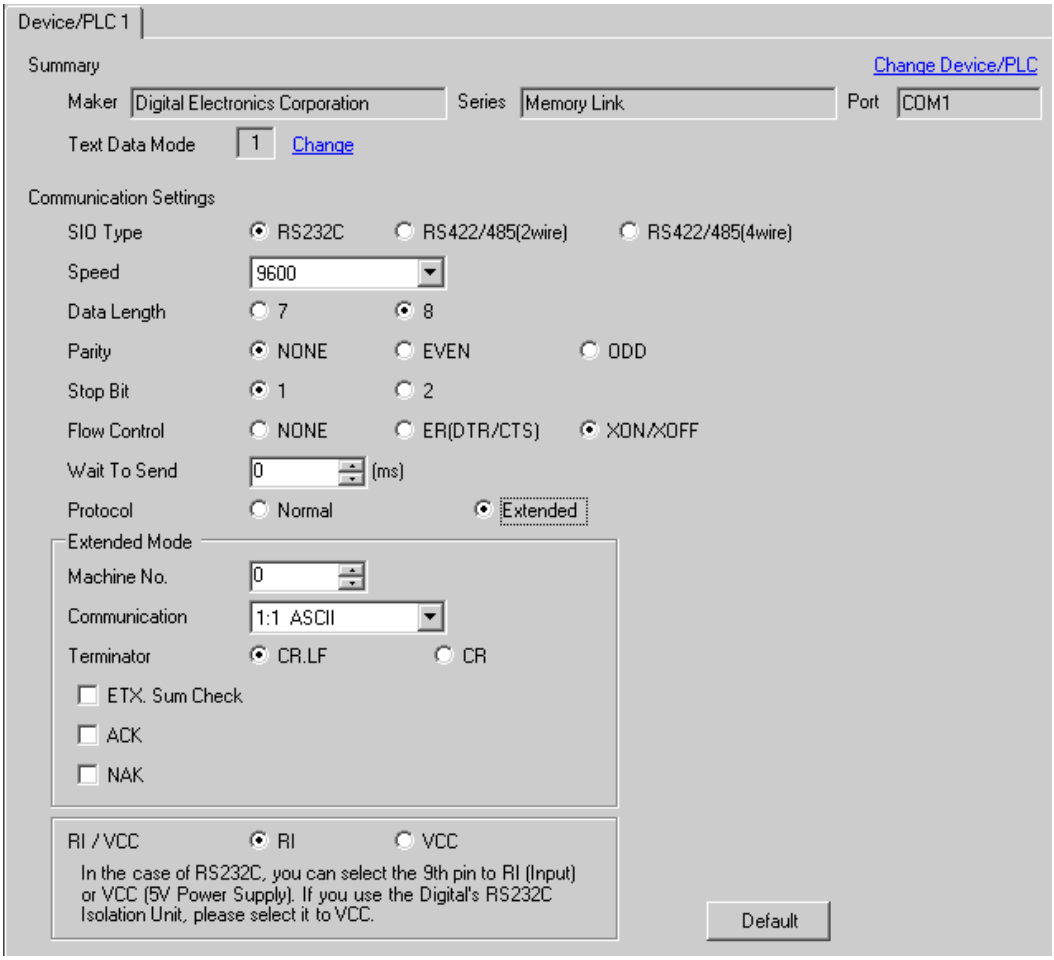
ตั้งค่าการสื่อสารของจอแสดงผลด้วย GP-Pro EX หรือตั้งค่าในโหมดออฟไลน์ของจอแสดงผล
ค่าของแต่ละพารามิเตอร์ต้องเหมือนกับค่าของอุปกรณ์ภายนอก

☞ “ตัวอย่างการตั้งค่าการสื่อสาร” (หน้า 48)

6.1 รายการตั้งค่าใน GP-Pro EX

■ การตั้งค่าการสื่อสาร (RS232C/RS422 (2wire) / RS422 (4wire))

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน

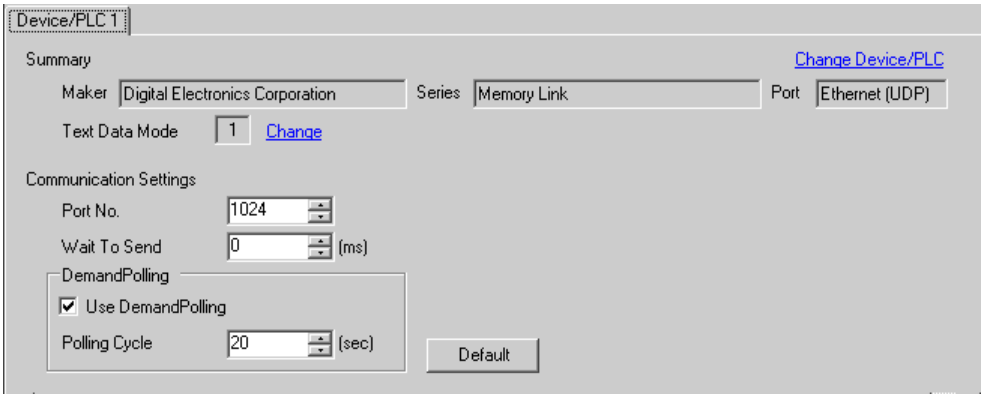


รายการตั้งค่า	คำอธิบายการตั้งค่า
SIO Type	เลือกชนิดของ SIO เพื่อสื่อสารกับอุปกรณ์ภายนอก
Speed	เลือกความเร็วในการสื่อสารระหว่างอุปกรณ์ภายนอกและจอแสดงผล
Data Length	เลือกความยาวข้อมูล
Parity	เลือกวิธีตรวจสอบพาริตี้
Stop Bit	เลือกความยาวของบิตสิ้นสุดการสื่อสาร

รายการตั้งค่า	คำอธิบายการตั้งค่า
Flow Control	เลือกวิธีการควบคุมการสื่อสารเพื่อป้องกันโอเวอร์โฟลว์ของข้อมูลการส่งและการรับ
Wait To Send	ป้อนระยะเวลาแสตนด์บาย (เป็นมิลลิวินาที) ของจอแสดงผลนับตั้งแต่รับแพ็กเก็ตจนถึงส่งคำสั่งครั้งต่อไป ด้วยจำนวนเต็มตั้งแต่ 0 ถึง 255
Protocol	เลือกโปรโตคอลการสื่อสาร
Machine No.	ป้อนหมายเลขยูนิตของอุปกรณ์ภายนอกที่จะใช้สื่อสาร ด้วยจำนวนเต็มตั้งแต่ 0 ถึง 31
Communication	เลือก SIO ชนิดใดชนิดหนึ่งระหว่าง “1:1 ASCII code”, “1:1 Binary code”, “1:N ASCII code” และ “1:N Binary code”
Terminator	เลือกเทอร์มิเนเตอร์ที่จะใช้
ETX. Sum Check	ตั้งค่าว่าจะเพิ่มรหัสการตรวจสอบผลรวมลงในการสื่อสารข้อมูลหรือไม่
ACK	เลือกตัวเลือกนี้หากคุณใช้ ACK
NAK	เลือกตัวเลือกนี้หากคุณใช้ NAK
RI/VCC	คุณสามารถสลับ RI/VCC ของขาที่ 9 ได้เมื่อคุณตั้งค่า SIO type เป็น RS232C

■ การตั้งค่าการสื่อสาร (Ethernet)

หากต้องการแสดงหน้าจอการตั้งค่า ให้เลือก [Device/PLC Settings] จาก [System setting window] ในพื้นที่ทำงาน



รายการตั้งค่า	คำอธิบายการตั้งค่า
Port No.	ป้อนหมายเลขพอร์ตของอุปกรณ์ภายนอกที่จะใช้สื่อสาร ด้วยจำนวนเต็มตั้งแต่ 1024 ถึง 65535
Wait To Send	ป้อนระยะเวลาแสตนด์บาย (เป็นมิลลิวินาที) ของจอแสดงผลนับตั้งแต่รับแพ็กเก็ตจนถึงส่งคำสั่งครั้งต่อไป ด้วยจำนวนเต็มตั้งแต่ 0 ถึง 255
Use Demand Polling	เลือกตัวเลือกนี้หากคุณใช้คำสั่งการโพลดีมานด์เพื่อยืนยันการมีอยู่ของอุปกรณ์ภายนอก
Polling Cycle	ป้อนรอบการโพล (เป็นวินาที) ด้วยจำนวนเต็มตั้งแต่ 10 ถึง 100

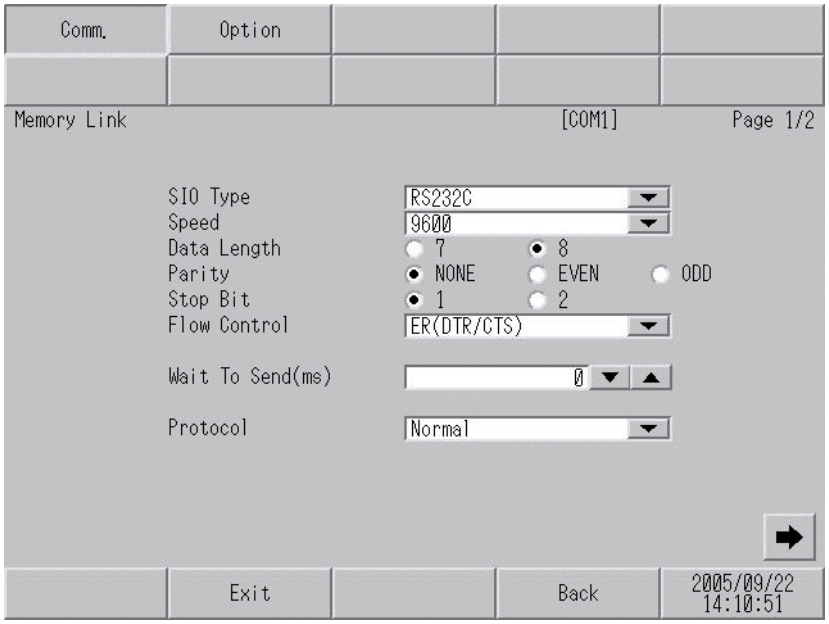
6.2 รายการตั้งค่าในโหมดออฟไลน์

หมายเหตุ
- โปรดดูข้อมูลเพิ่มเติมเกี่ยวกับวิธีเข้าสู่โหมดออฟไลน์หรือข้อมูลการดำเนินการได้จากคู่มือผู้ใช้สำหรับ GP3000 Series
  Cf. คู่มือผู้ใช้สำหรับ GP3000 Series “บทที่ 4 การตั้งค่า”

■ การตั้งค่าของ RS232C/RS422 (2wire) / RS422 (4wire)

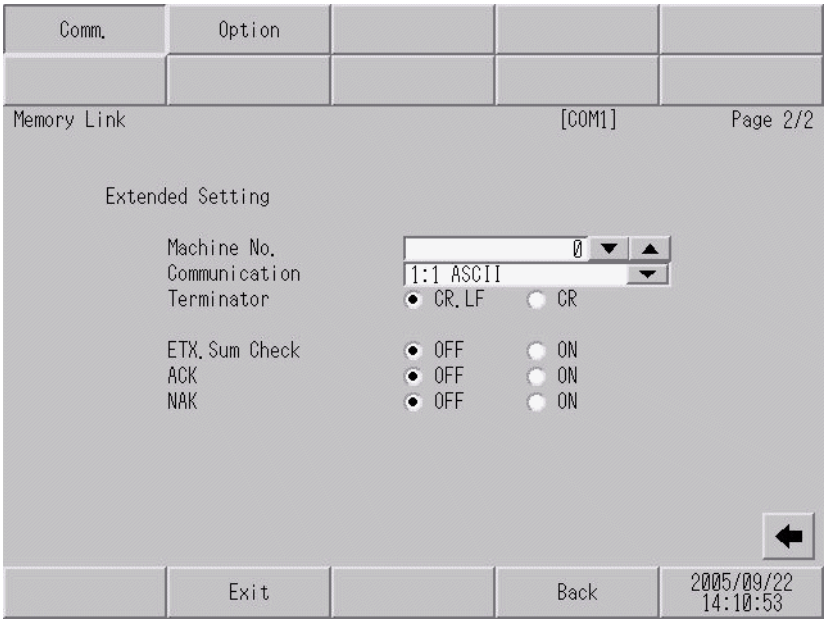
◆ การตั้งค่าการสื่อสาร

หากต้องการแสดงหน้าจอการตั้งค่า ให้แตะ [Device/PLC Settings] จาก [Peripheral Settings] ในโหมดออฟไลน์ จากนั้นแตะที่อุปกรณ์ภายนอกที่คุณต้องการตั้งค่าจากรายชื่อที่แสดงอยู่ แล้วแตะที่ [Communication Settings] (หน้า 1/2)



รายการตั้งค่า	คำอธิบายการตั้งค่า
SIO Type	เลือกชนิดของ SIO เพื่อสื่อสารกับอุปกรณ์ภายนอก
Speed	เลือกความเร็ว (bps) ในการสื่อสารระหว่างอุปกรณ์ภายนอกและจอแสดงผล
Data Length	เลือกความยาวข้อมูล
Parity	เลือกวิธีตรวจสอบพาริตี้
Stop Bit	เลือกความยาวของบิตสิ้นสุดการสื่อสาร
Flow Control	เลือกวิธีการควบคุมการสื่อสารเพื่อป้องกันโอเวอร์โฟลว์ของข้อมูลการส่งและการรับ
Wait To Send	ป้อนระยะเวลาแสตนด์บาย (เป็นมิลลิวินาที) ของจอแสดงผลนับตั้งแต่รับแพ็กเก็ตจนถึงส่งคำสั่งครั้งต่อไป ด้วยจำนวนเต็มตั้งแต่ 0 ถึง 255
Protocol	เลือกโปรโตคอลการสื่อสารระหว่าง “Normal” หรือ “Extended”

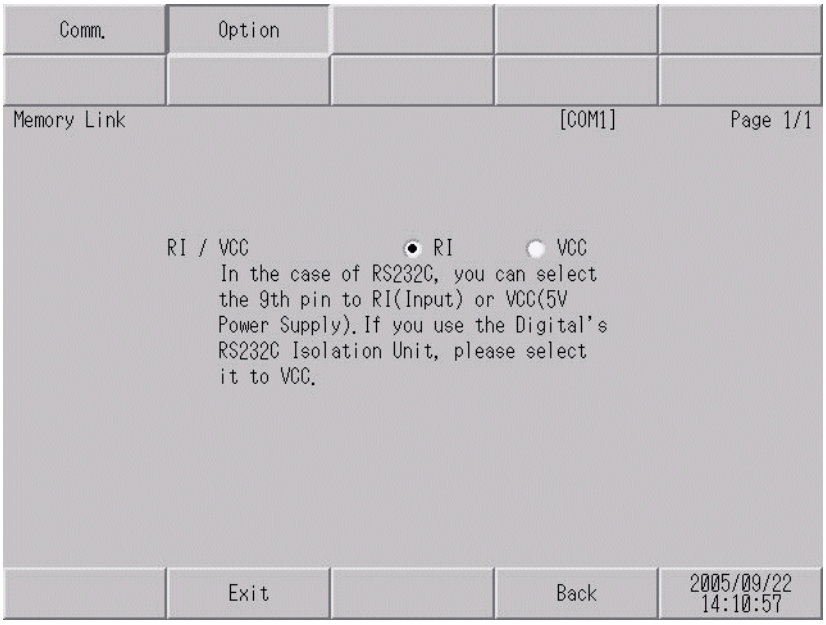
(หน้า 2/2)



รายการตั้งค่า	คำอธิบายการตั้งค่า
Machine No.	ป้อนหมายเลขยูนิตของอุปกรณ์ภายนอกที่จะใช้สื่อสาร ด้วยจำนวนเต็มตั้งแต่ 0 ถึง 31
Communication	เลือก SIO ชนิดใดชนิดหนึ่งระหว่าง “1:1 ASCII code”, “1:1 Binary code”, “1:N ASCII code” และ “1:N Binary code”
Terminator	เลือกเทอร์มิเนเตอร์ที่จะใช้ระหว่าง “CR, LF” หรือ “CR“
ETX. Sum Check	ตั้งค่าว่าจะเพิ่มรหัสการตรวจสอบผลรวมลงในการสื่อสารข้อมูลหรือไม่
ACK	เลือกตัวเลือกนี้หากคุณใช้ ACK
NAK	เลือกตัวเลือกนี้หากคุณใช้ NAK

◆ **ตัวเลือก**

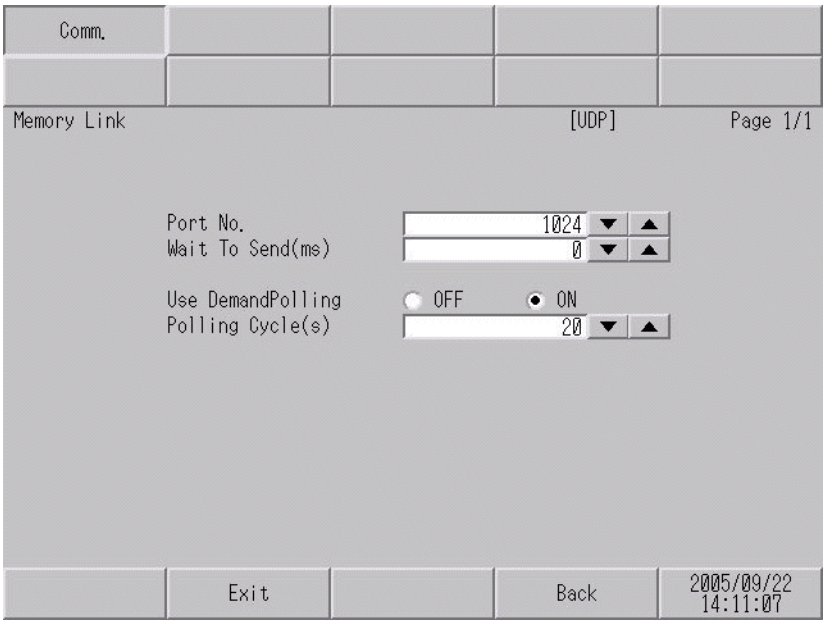
หากต้องการแสดงหน้าจอการตั้งค่า ให้แตะ [Device/PLC Settings] จาก [Peripheral Settings] ในโหมดออฟไลน์ จากนั้นแตะที่อุปกรณ์ภายนอกที่คุณต้องการตั้งค่าจากรายการที่แสดงอยู่ แล้วแตะที่ [Option]



รายการตั้งค่า	คำอธิบายการตั้งค่า
RI/VCC	คุณสามารถสลับ RI/VCC ของขาที่ 9 ได้เมื่อคุณตั้งค่า SIO type เป็น RS232C

■ การตั้งค่าของ Ethernet

หากต้องการแสดงหน้าจอการตั้งค่า ให้แตะ [Device/PLC Settings] จาก [Peripheral Settings] ในโหมดออฟไลน์ จากนั้นแตะที่อุปกรณ์ภายนอกที่คุณต้องการตั้งค่าจากรายชื่อที่แสดงอยู่ แล้วแตะที่ [Communication Settings]



รายการตั้งค่า	คำอธิบายการตั้งค่า
Port No.	ป้อนหมายเลขพอร์ตของอุปกรณ์ภายนอกที่จะใช้สื่อสาร ด้วยจำนวนเต็มตั้งแต่ 1024 ถึง 65535
Wait To Send	ป้อนระยะเวลาแสตนด์บาย (เป็นมิลลิวินาที) ของจอแสดงผลนับตั้งแต่รับแพ็กเก็ตจนถึงส่งคำสั่งครั้งต่อไป ด้วยจำนวนเต็มตั้งแต่ 0 ถึง 255
Use Demand Polling	เลือกตัวเลือกนี้หากคุณใช้คำสั่งการโพลดีมานด์เพื่อยืนยันการมีอยู่ของอุปกรณ์ภายนอก
Polling Cycle (s)	ป้อนรอบการโพล (เป็นวินาที) ด้วยจำนวนเต็มตั้งแต่ 10 ถึง 100

7 แผนภาพแสดงการต่อสายเคเบิล

แผนภาพแสดงการต่อสายเคเบิลที่แสดงอยู่ที่ด้านล่างนี้อาจแตกต่างไปจากแผนภาพแสดงการต่อสายเคเบิลที่ผู้ผลิตอุปกรณ์ภายนอกแนะนำให้ใช้ โปรดตรวจสอบให้แน่ใจว่าการปฏิบัติตามแผนภาพแสดงการต่อสายเคเบิลซึ่งแสดงไว้ในคู่มือนี้ไม่ทำให้เกิดปัญหาในการปฏิบัติงาน

- **ขา FG ของตัวเครื่องหลักของอุปกรณ์ภายนอกจะต้องต่อลงดินแบบ D-Class โปรดดูรายละเอียดเพิ่มเติมจากคู่มือของอุปกรณ์ภายนอก**
- **SG และ FG เชื่อมต่อกันภายในจอแสดงผล เมื่อเชื่อมต่อ SG กับอุปกรณ์ภายนอก ให้ออกแบบระบบไม่ให้เกิดการลัดวงจร**

แผนภาพแสดงการต่อสายเคเบิล 1

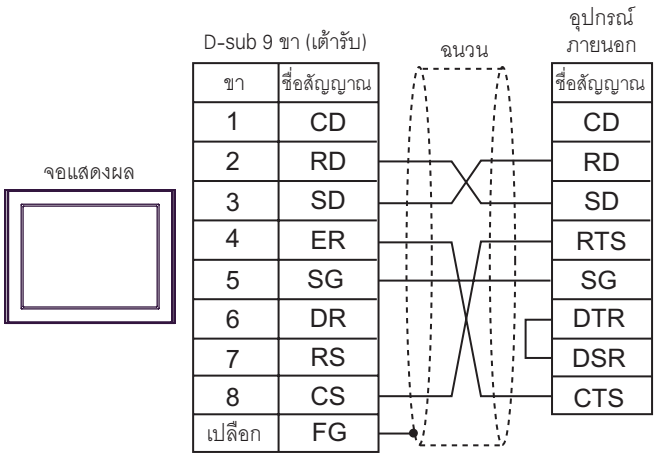
จอแสดงผล (พอร์ตเชื่อมต่อ)	สายเคเบิล		หมายเหตุ
GP (COM1)	A	สายเคเบิลของคุณเอง (การควบคุม ER)	สายเคเบิลต้องยาวไม่เกิน 15 เมตร
	B	สายเคเบิลของคุณเอง (การควบคุม X)	
	C	สายเคเบิลของคุณเอง (ไม่มีวิธีการควบคุม)	

ข้อสำคัญ

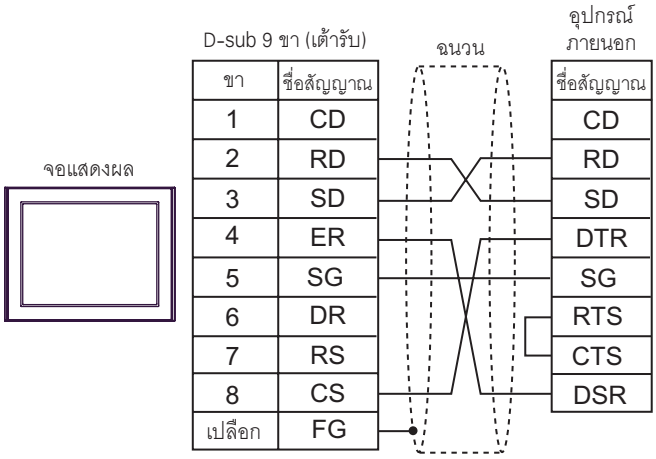
- การเชื่อมต่อคอนเนคเตอร์ชนิด RS232C หรือหมายเลขขาให้ตรงกับชื่อสัญญาณจะแตกต่างกันไปตามอุปกรณ์โฮสต์ โปรดเชื่อมต่อให้ถูกต้องตามข้อมูลจำเพาะของอินเตอร์เฟซของอุปกรณ์โฮสต์

A) **เมื่อใช้สายเคเบิลของคุณเอง**

- **เมื่ออุปกรณ์ภายนอกรองรับการควบคุมแบบ RTS/CTS**



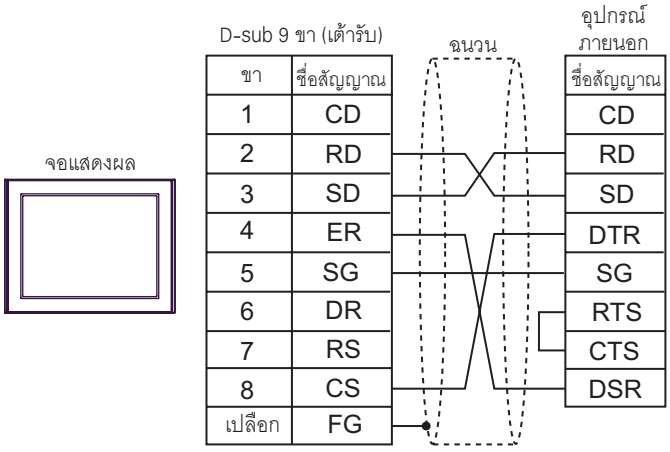
- **เมื่ออุปกรณ์ภายนอกรองรับการควบคุมแบบ DTR/DSR**



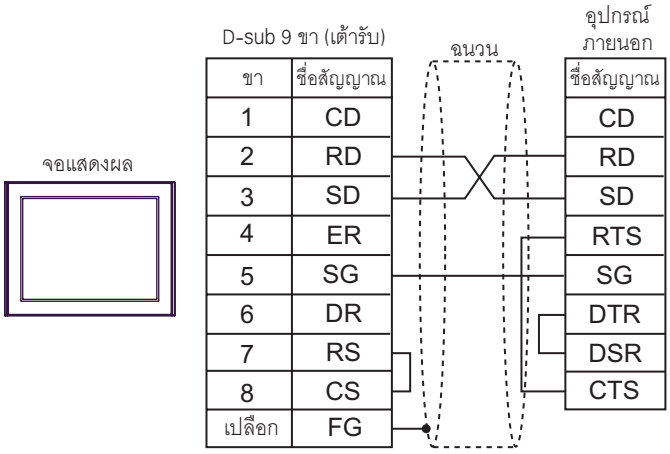
ข้อห้าม:

- **ขณะที่ตัวแปร ER ใน GP ปิดอยู่ ห้ามให้อุปกรณ์โฮสต์ส่งข้อมูล**

B) **เมื่อใช้สายเคเบิลของคุณเอง (การควบคุม X)**



C) เมื่อใช้สายเคเบิลของคุณเอง (ไม่มีวิธีการควบคุม)



แผนภาพแสดงการต่อสายเคเบิล 2

จอแสดงผล (พอร์ตเชื่อมต่อ)	สายเคเบิล		หมายเหตุ
GP*1(COM1) AGP-3302 (COM2)	A	ตัวแปลงพอร์ตสื่อสาร ของ Pro-face CA3-ADPCOM-01 + ตัวแปลงสำหรับต่อกับเทอร์มินัลบล็อคของคอนเนคเตอร์ ของ Pro-face CA3-ADPTRM-01 + สายเคเบิลของคุณเอง	
	B	ตัวแปลงพอร์ตสื่อสาร ของ Pro-face CA3-ADPCOM-01 + สายเคเบิล 422 สำหรับ AGP ของ Pro-face CA3-CBL422-01	
	C	ตัวแปลงพอร์ตสื่อสาร ของ Pro-face CA3-ADPCOM-01 + สายเคเบิลมัลติลิงค์สำหรับ AGP ของ Pro-face CA3-CBLMLT-01	
	D	สายเคเบิลของคุณเอง	
GP*1 (COM2)	E	อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ของ Pro-face CA4-ADPONL-01 + ตัวแปลงสำหรับต่อกับเทอร์มินัลบล็อคของคอนเนคเตอร์ ของ Pro-face CA3-ADPTRM-01 + สายเคเบิลของคุณเอง	
	F	อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ของ Pro-face CA4-ADPONL-01 + สายเคเบิล 422 สำหรับ AGP ของ Pro-face CA3-CBL422-01	
	G	อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ของ Pro-face CA4-ADPONL-01 + สายเคเบิลมัลติลิงค์สำหรับ AGP ของ Pro-face CA3-CBLMLT-01	
	H	อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ของ Pro-face CA4-ADPONL-01 + สายเคเบิลของคุณเอง	

*1 GP ทุกรุ่นยกเว้น AGP3302 Series

หมายเหตุ • วิธีควบคุมเมื่อใช้สายเคเบิล RS422 คือ XON/XOFF เท่านั้น การควบคุม XON/XOFF ใช้ได้เฉพาะสำหรับ ASCII

ข้อบังคับ:

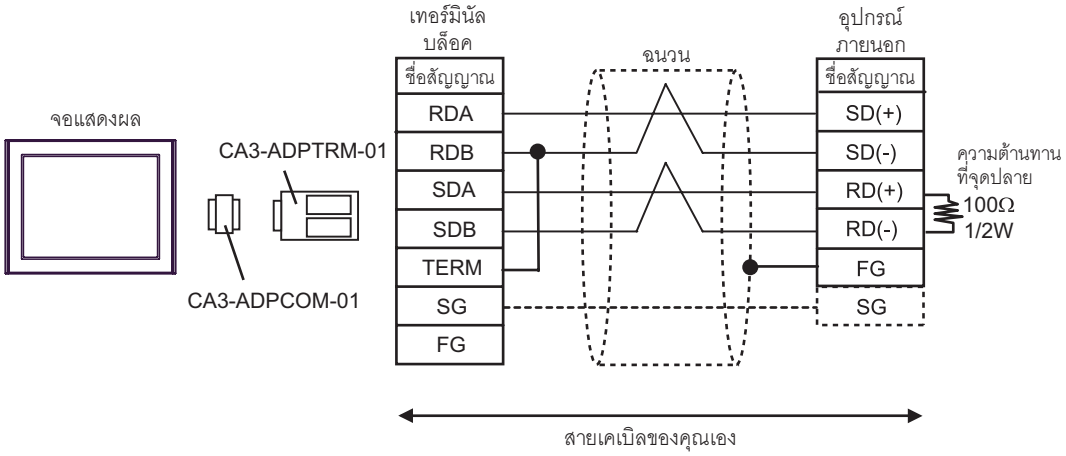
- **ใช้สายเคเบิลตีเกลียวที่มีความจุประมาณ 50pF/m, ความต้านทาน 100 Ω, ทำจากลวดขนาด 24AWG**

ข้อสำคัญ

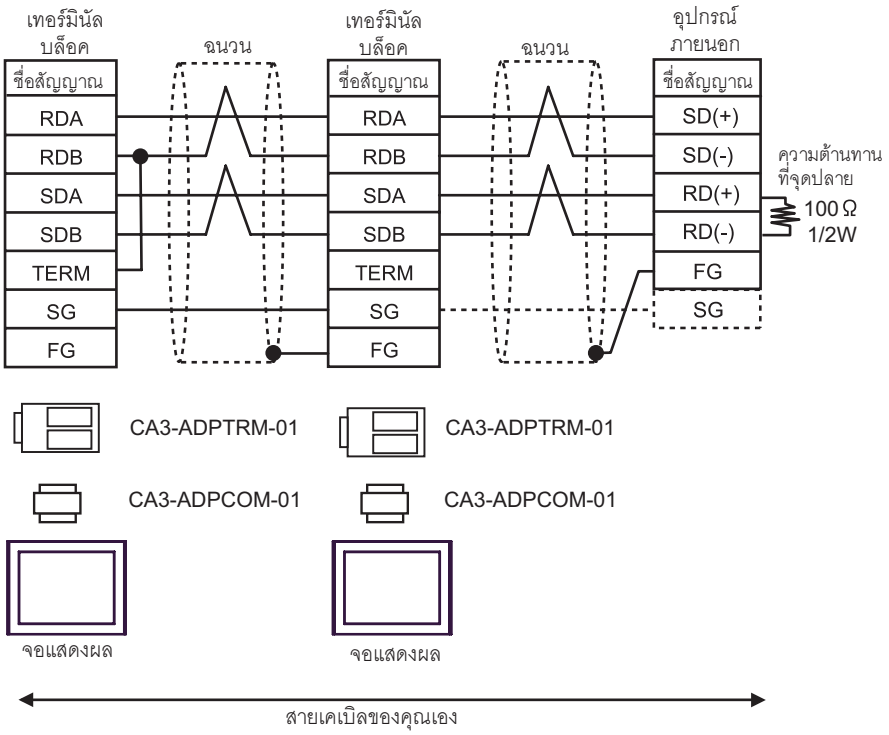
- โดยปกติสายเคเบิล RS422 จะมีความยาวสูงสุด 1000 ม. แต่ความยาวสายจะถูกจำกัดตามอุปกรณ์โฮสต์ที่เชื่อมต่อ ในการเชื่อมต่อ โปรดดูข้อมูลในคู่มือของอุปกรณ์โฮสต์ที่เชื่อมต่อ
- วิธีการเชื่อมต่อหรือความต้านทานที่จุดปลายจะแตกต่างกันไปตามอุปกรณ์โฮสต์ที่เชื่อมต่อ
- ไม่แยกที่ฝั่ง GP
- ให้ต่อขา SG ระหว่างเครื่อง GP หลายเครื่องเสมอ
- หากอุปกรณ์ภายนอกถูกแยกออก ไม่จำเป็นต้องมีการเชื่อมต่อขา SG ระหว่างอุปกรณ์ภายนอกกับ GP

A) เมื่อใช้ตัวแปลงพอร์ตสื่อสาร (CA3-ADPCOM-01), ตัวแปลงสำหรับต่อกับเทอร์มินัลบล็อคของคอนเนคเตอร์ (CA3-ADPTRM-01) ของ Pro-face และสายเคเบิลของคุณเอง

- **การเชื่อมต่อแบบ 1:1**



การเชื่อมต่อแบบ 1:n

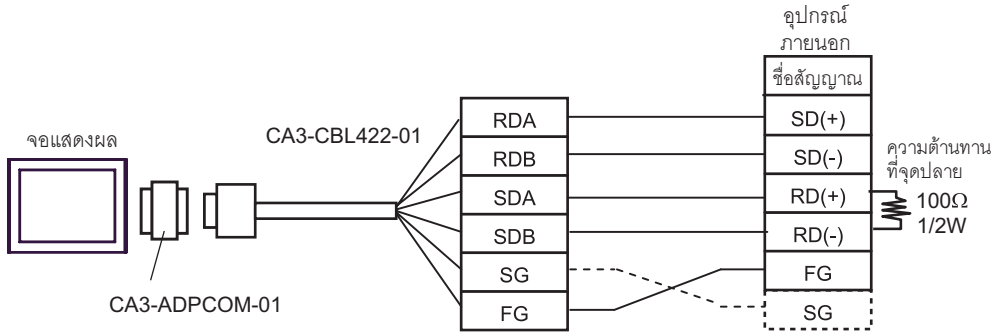


หมายเหตุ

- เชื่อมต่อ RDB ของ CA3-ADPTRM-01 เข้ากับ TERM เพื่อแทรกความต้านทานที่จุดปลายขนาด 100Ω 1/2W ในระหว่าง RDA กับ RDB ที่ด้านเครื่อง GP

B) เมื่อใช้ตัวแปลงพอร์ตสื่อสาร (CA3-ADPCOM-01), สายเคเบิล 422 สำหรับ AGP (CA3-CBL422-01) ของ Pro-face และสายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:1

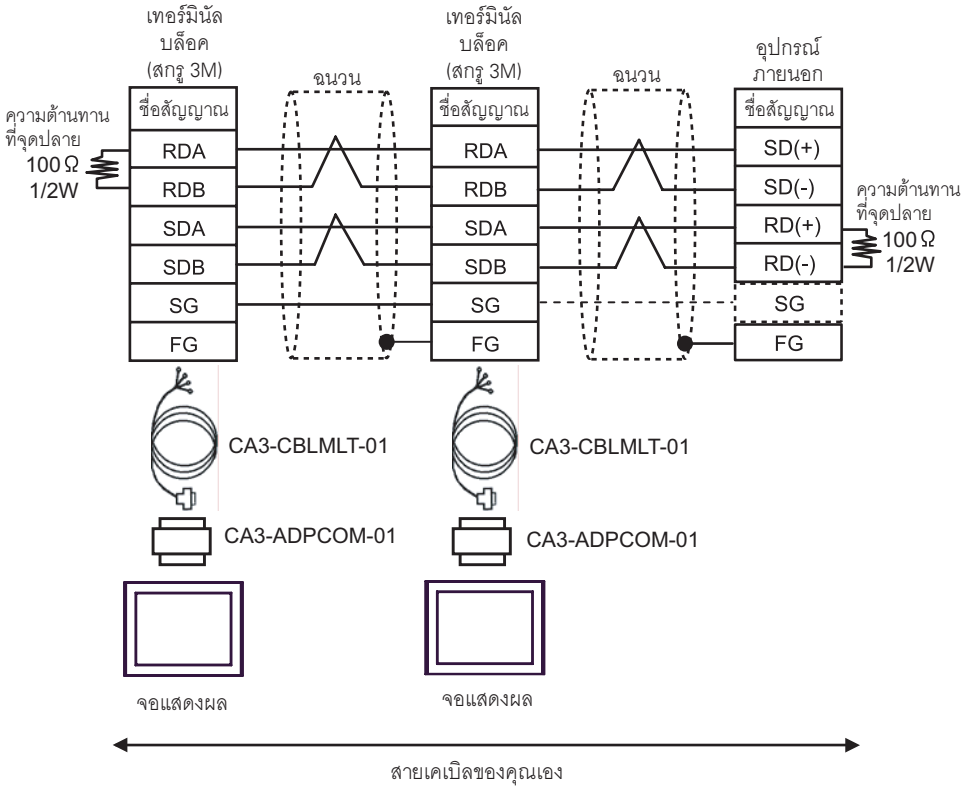


หมายเหตุ

- มีการแทรกความต้านทานที่จุดปลายขนาด 100Ω ระหว่าง RDA กับ RDB ใน CA3-CBL422-01

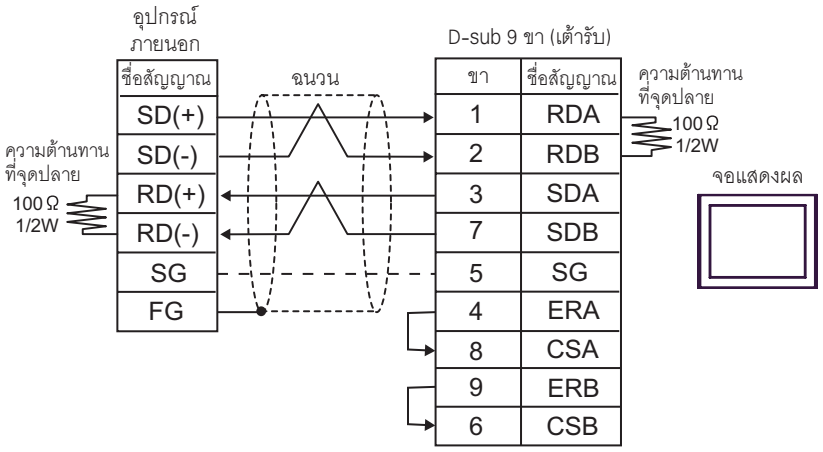
C) เมื่อใช้ตัวแปลงพอร์ตสื่อสาร (CA3-ADPCOM-01), สายเคเบิลมัลติลิงค์สำหรับ AGP (CA3-CBLMLT-01) ของ Pro-face และสายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:n

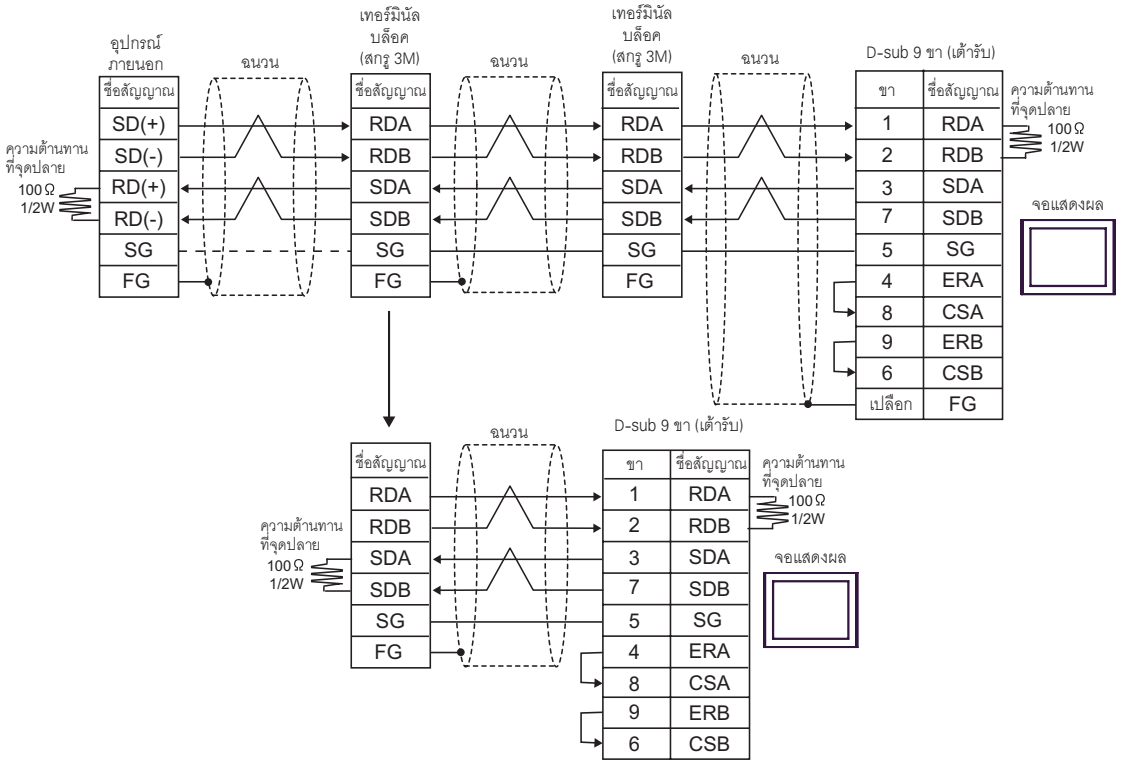


D) เมื่อใช้สายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:1

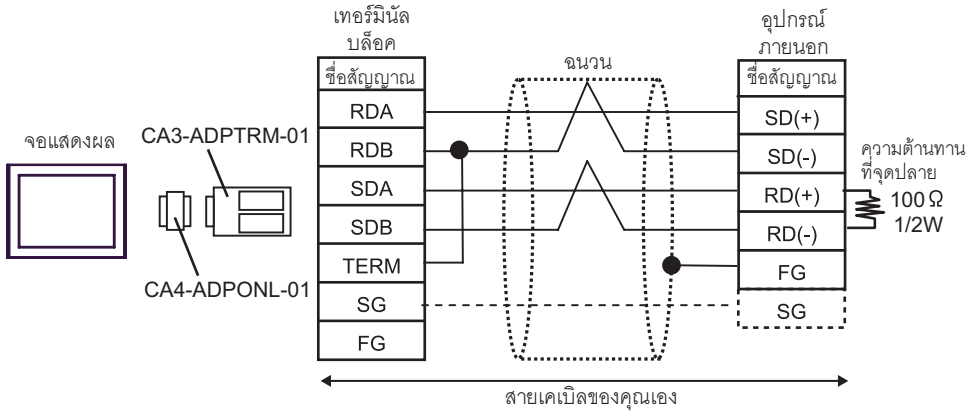


- **การเชื่อมต่อแบบ 1:n**

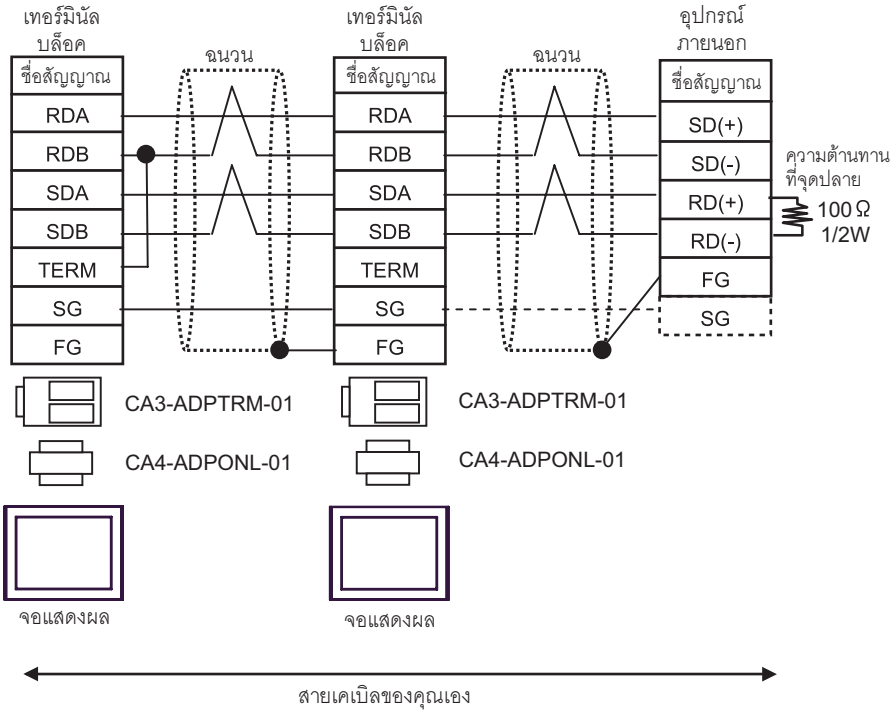


E) **เมื่อใช้อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ (CA4-ADPONL-01), ตัวแปลงสำหรับต่อกับเทอร์มินัลบล็อคของคอนเนคเตอร์ (CA3-ADPTRM-01) ของ Pro-face และสายเคเบิลของคุณเอง**

- **การเชื่อมต่อแบบ 1:1**



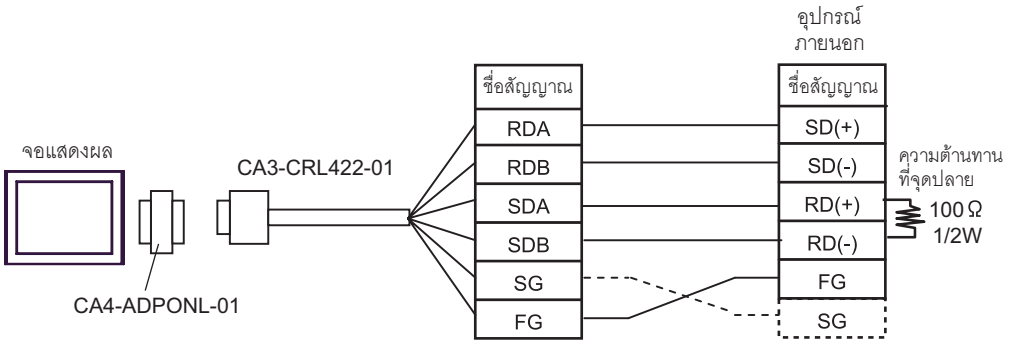
- **การเชื่อมต่อแบบ 1:n**



หมายเหตุ	• เชื่อมต่อ RDA ของ CA3-ADPTRM-01 เข้ากับ TERM เพื่อแทรกความต้านทานที่จุดปลายขนาด 100Ω 1/2W ระหว่าง RDA กับ RDB ที่ฝั่ง GP

F) เมื่อใช้อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ (CA4-ADPONL-01), สายเคเบิล 422 สำหรับ AGP (CA3-CBL422-01) ของ Pro-face

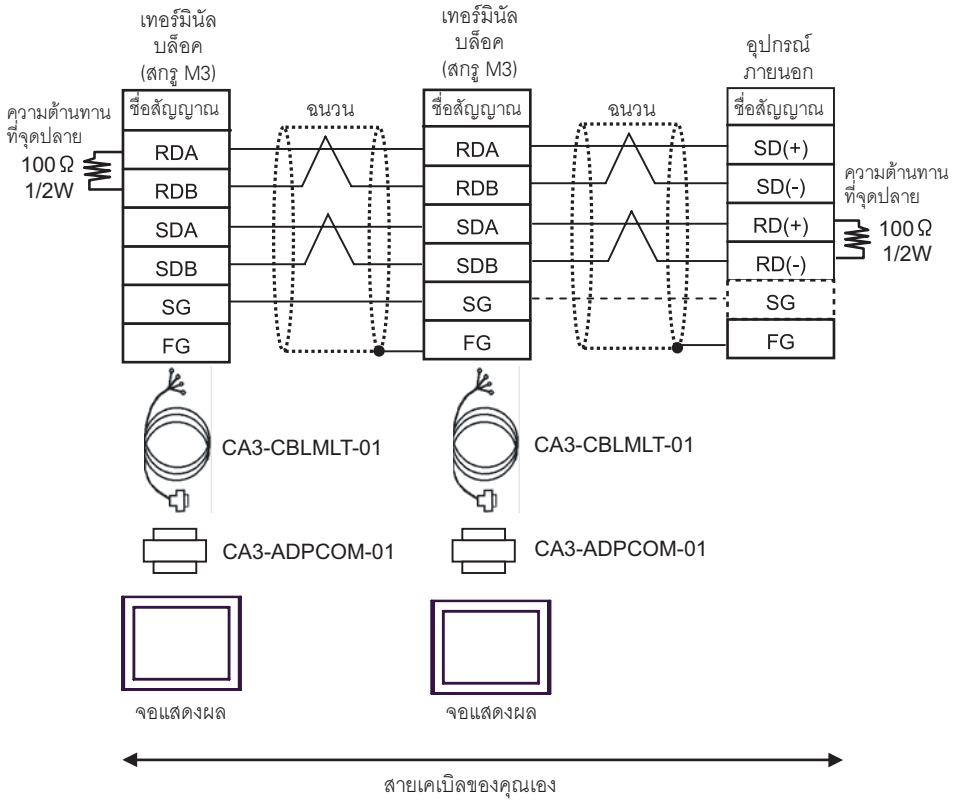
- **การเชื่อมต่อแบบ 1:1**



หมายเหตุ	• มีการแทรกความต้านทานที่จุดปลายขนาด 100Ω ระหว่าง RDA กับ RDB ใน CA3-CBL422-01

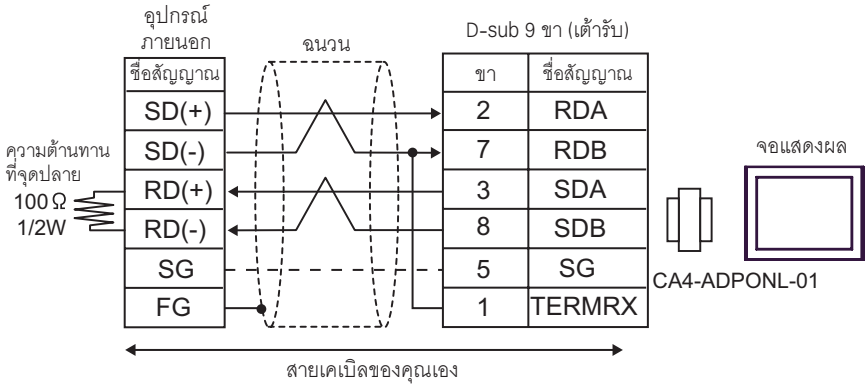
G) เมื่อใช้อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ (CA4-ADPONL-01) และสายเคเบิลมัลติลิงค์สำหรับ AGP (CA3-CBLMLT-01) ของ Pro-face และสายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:n

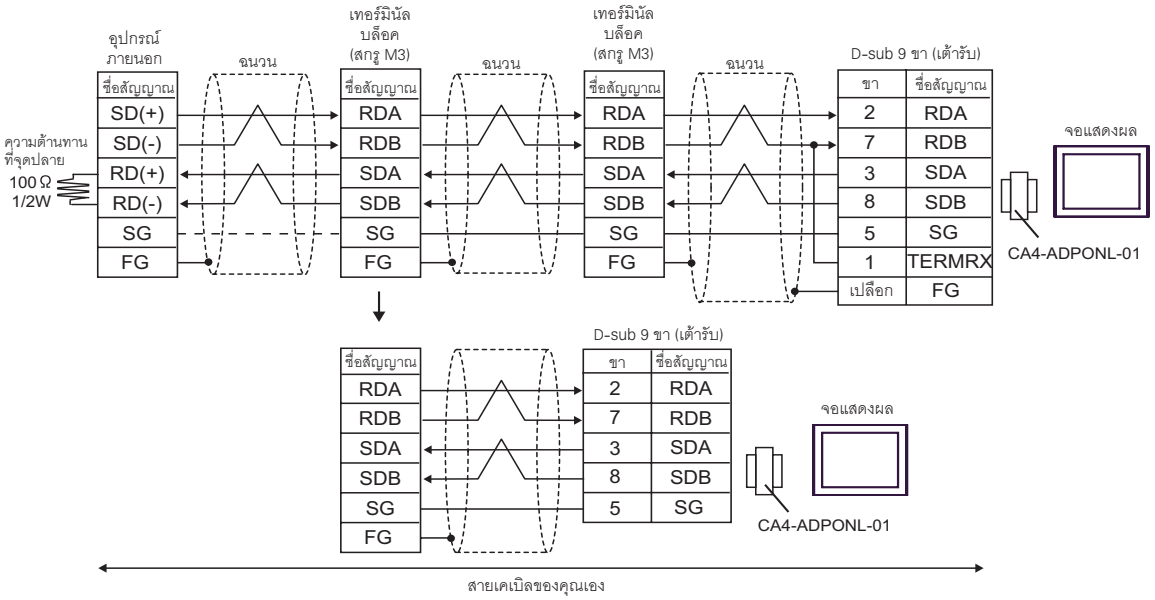


H) เมื่อใช้อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ (CA4-ADPONL-01) ของ Pro-face และสายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:1



- **การเชื่อมต่อแบบ 1:n**



แผนภาพแสดงการต่อสายเคเบิล 3 (การเชื่อมต่อแบบ RS-422 (2 wire))

Display (พอร์ตเชื่อมต่อ)	สายเคเบิล		หมายเหตุ
GP[*1](COM1) AGP-3302 (COM2)	A	ตัวแปลงพอร์ตสื่อสาร ของ Pro-face CA3-ADPCOM-01 + ตัวแปลงสำหรับต่อกับเทอร์มินัลบล็อคของคอนเนคเตอร์ ของ Pro-face CA3-ADPTRM-01 + สายเคเบิลของคุณเอง	
	B	สายเคเบิลของคุณเอง	
GP*1 (COM2)	C	อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ของ Pro-face CA4-ADPONL-01 + ตัวแปลงสำหรับต่อกับเทอร์มินัลบล็อคของคอนเนคเตอร์ ของ Pro-face CA3-ADPTRM-01 + สายเคเบิลของคุณเอง	
	D	อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ของ Pro-face CA4-ADPONL-01 + สายเคเบิลของคุณเอง	

*1 GP ทุกรุ่นยกเว้น AGP-3302

หมายเหตุ
- วิธีควบคุมเมื่อใช้สายเคเบิล RS422 คือ XON/XOFF เท่านั้น การควบคุม XON/XOFF ใช้ได้เฉพาะสำหรับ ASCII

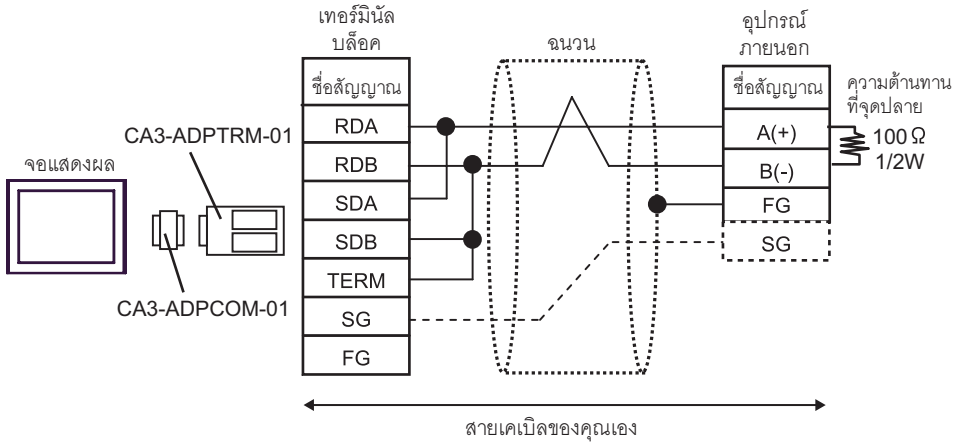
ข้อบังคับ:

- **ใช้สายเคเบิลตีเกลียวที่มีความจุประมาณ 50pF/m, ความต้านทาน 100Ω, ทำจากลวดขนาด 24AWG**

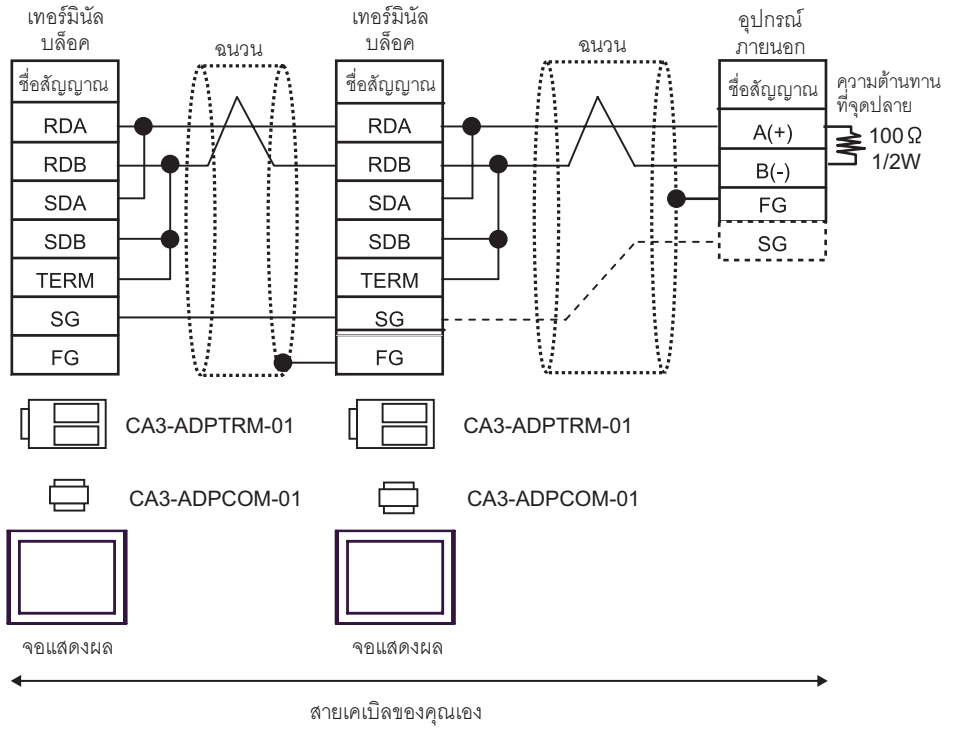
ข้อสำคัญ
- โดยปกติสายเคเบิล RS422 จะมีความยาวสูงสุด 1000 ม. แต่ความยาวสายจะถูกจำกัดตามอุปกรณ์โฮสต์ที่เชื่อมต่อ ในการเชื่อมต่อ โปรดดูข้อมูลในคู่มือของอุปกรณ์โฮสต์ที่เชื่อมต่อ
- วิธีการเชื่อมต่อหรือความต้านทานที่จุดปลายจะแตกต่างกันไปตามอุปกรณ์โฮสต์ที่เชื่อมต่อ หากมี SG ให้เชื่อมต่อด้วย

A) เมื่อใช้ตัวแปลงพอร์ตสื่อสาร (CA3-ADPCOM-01), ตัวแปลงสำหรับต่อกับเทอร์มินัลบล็อคของคอนเนคเตอร์ (CA3-ADPTRM-01) ของ Pro-face และสายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:1



- การเชื่อมต่อแบบ n:1

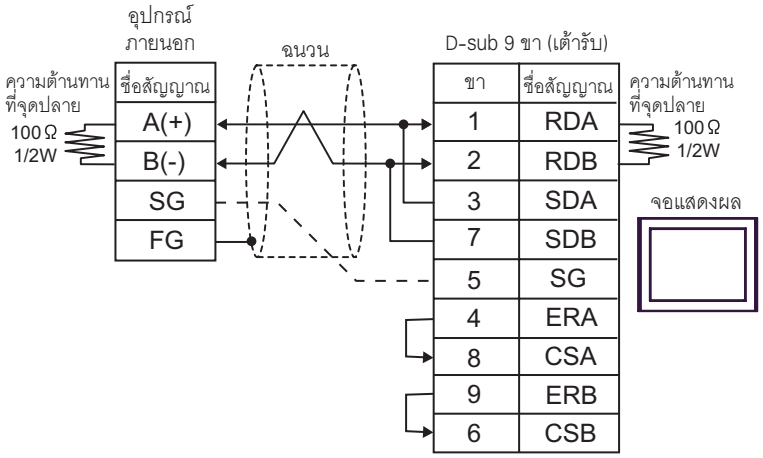


หมายเหตุ

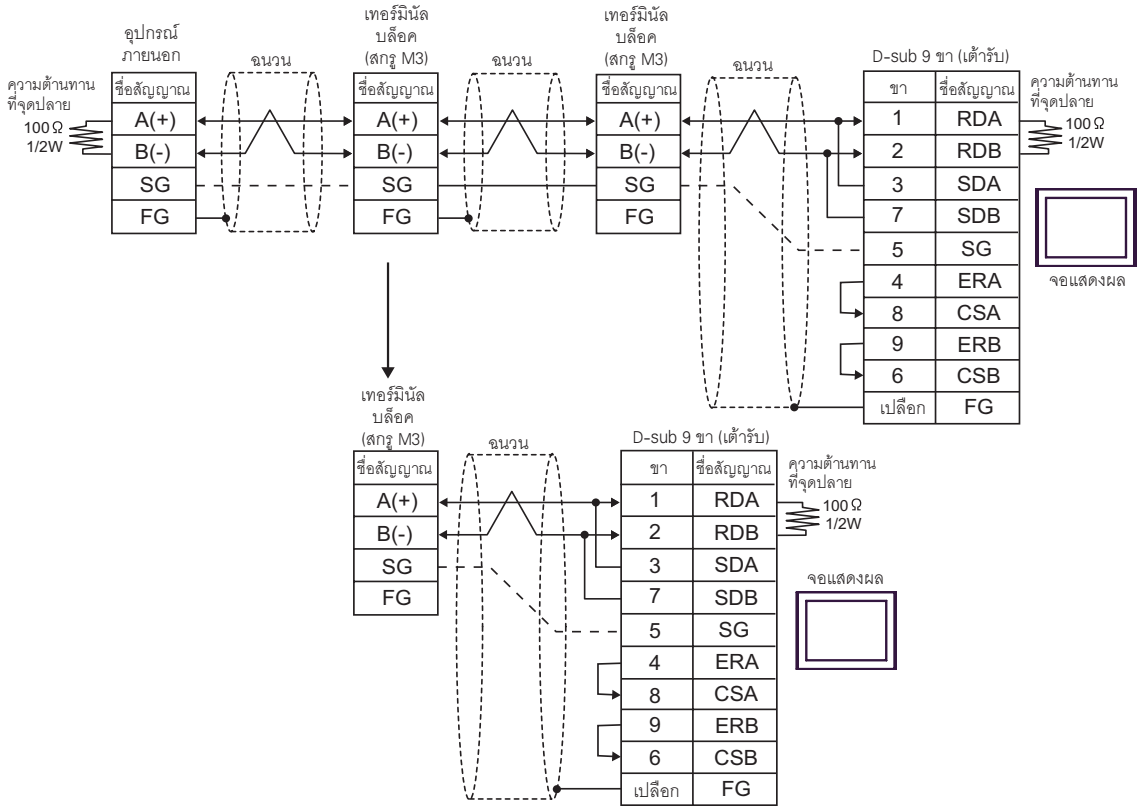
- เชื่อมต่อ RDB ของ CA3-ADPTRM-01 เข้ากับ TERM เพื่อแทรกความต้านทานที่จุดปลายขนาด 100Ω 1/2W ระหว่าง RDA กับ RDB ที่จอแสดงผล

B) เมื่อใช้สายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:1

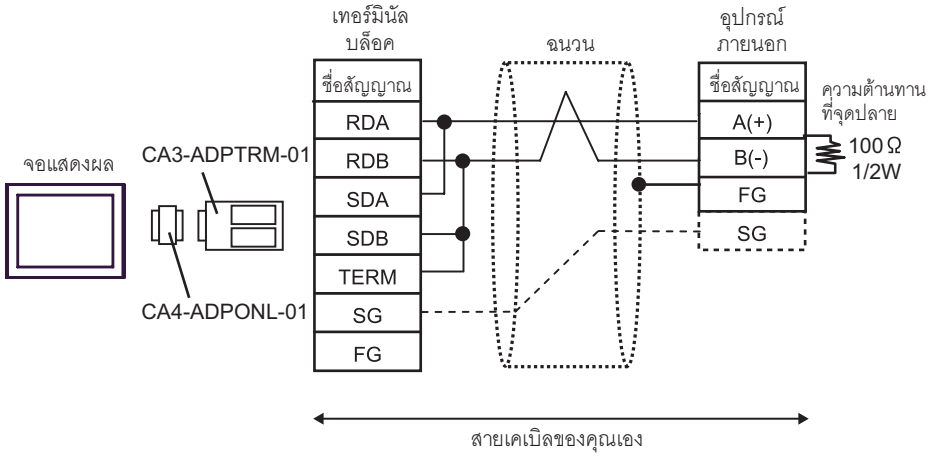


- การเชื่อมต่อแบบ n:1

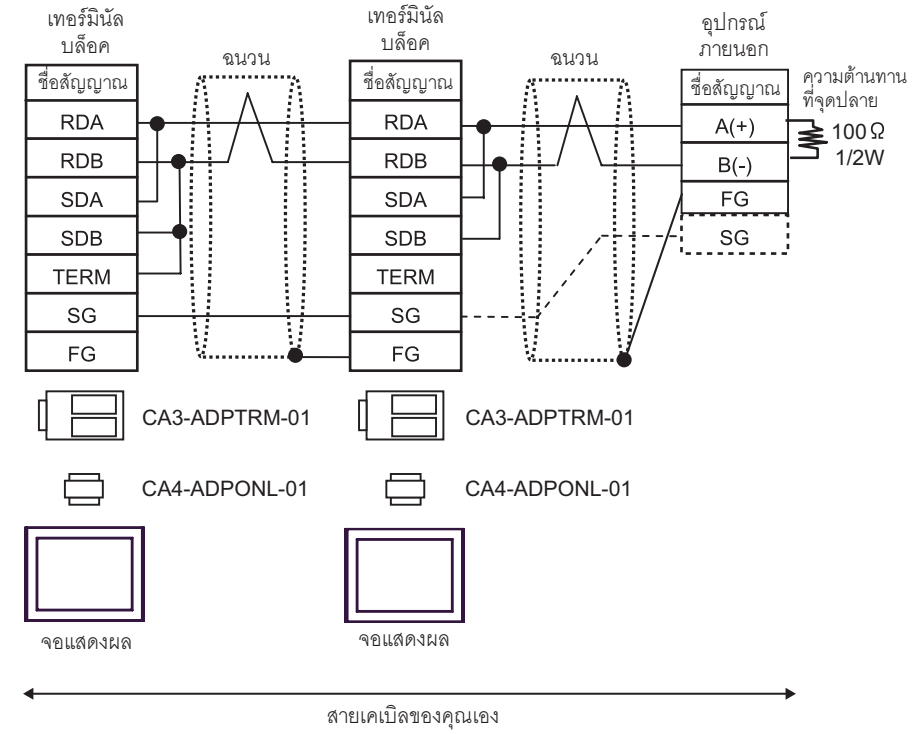


C) เมื่อใช้อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ (CA4-ADPONL-01), ตัวแปลงสำหรับต่อกับเทอร์มินัลบล็อคของคอนเนคเตอร์ (CA3-ADPTRM-01) ของ Pro-face และสายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:1

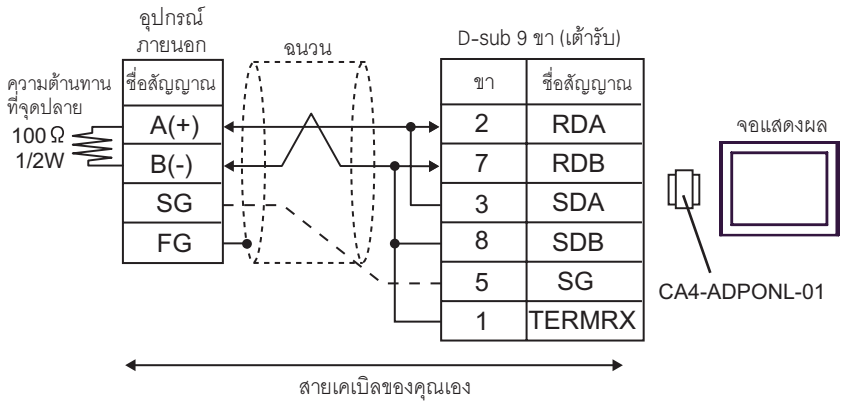


- การเชื่อมต่อแบบ n:1

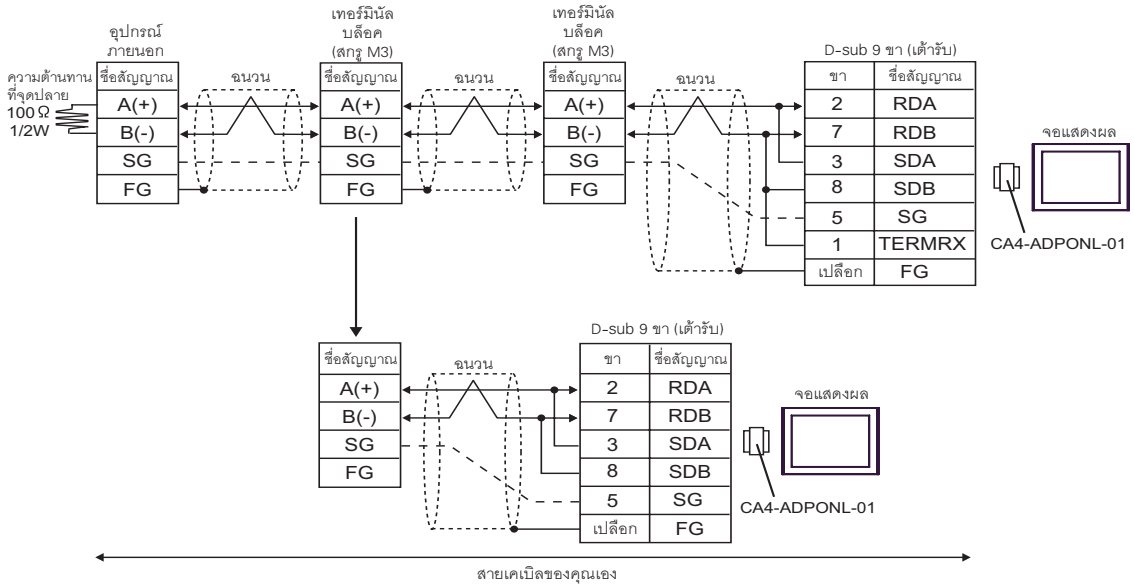


D) เมื่อใช้อะแดปเตอร์สำหรับเชื่อมต่อออนไลน์ (CA4-ADPONL-01) ของ Pro-face และสายเคเบิลของคุณเอง

- การเชื่อมต่อแบบ 1:1



- การเชื่อมต่อแบบ n:1



8 อุปกรณ์ที่รองรับ

ตารางด้านล่างนี้แสดงช่วงตำแหน่งอุปกรณ์ที่รองรับ โปรดทราบว่าช่วงของอุปกรณ์ที่รองรับจริงจะแตกต่างกันไป โดยขึ้นอยู่กับอุปกรณ์ภายนอกที่จะใช้ โปรดตรวจสอบช่วงจริงในคู่มือของอุปกรณ์ภายนอกของคุณ

อุปกรณ์	ตำแหน่งบิต	ตำแหน่งเวิร์ด	32 บิต	หมายเหตุ
อุปกรณ์ภายใน	000000 – 999915	0000 – 9999	H/L	

หมายเหตุ
- โปรดดูรายละเอียดเกี่ยวกับไอคอนในตารางจากข้อควรระวังในคู่มือ
  ☞ "สัญลักษณ์และคำศัพท์ที่ใช้ในคู่มือ"

9 รหัสอุปกรณ์และรหัสตำแหน่ง

ใช้รหัสอุปกรณ์และรหัสตำแหน่งเมื่อคุณเลือกชนิดตำแหน่งของการแสดงข้อมูลเป็น “Device Type & Address”

อุปกรณ์	ชื่ออุปกรณ์	รหัสอุปกรณ์ (HEX)	รหัสตำแหน่ง
อุปกรณ์ภายใน	-	0000	ค่าของตำแหน่งเวิร์ด

10 ข้อความแสดงข้อผิดพลาด

หน้าจอแสดงผลจะแสดงข้อความแสดงข้อผิดพลาดในรูปแบบต่อไปนี้ “หมายเลข: ชื่ออุปกรณ์: ข้อความแสดงข้อผิดพลาด (พื้นที่ที่เกิดข้อผิดพลาด)” คุณสามารถดูคำอธิบายของแต่ละรายการได้ที่ด้านล่าง

รายการ	คำอธิบาย
หมายเลข	หมายเลขข้อผิดพลาด
ชื่ออุปกรณ์	ชื่อของอุปกรณ์ภายนอกที่เกิดข้อผิดพลาด ชื่ออุปกรณ์คือชื่อของอุปกรณ์ภายนอกที่ตั้งค่าด้วย GP-Pro EX (ค่าเริ่มต้นคือ [PLC1])
ข้อความแสดงข้อผิดพลาด	แสดงข้อความที่เกี่ยวกับข้อผิดพลาดที่เกิดขึ้น
พื้นที่ที่เกิดข้อผิดพลาด	แสดงตำแหน่ง IP หรือตำแหน่งอุปกรณ์ของอุปกรณ์ภายนอกที่เกิดข้อผิดพลาด หรือรหัสข้อผิดพลาดที่ได้รับจากอุปกรณ์ภายนอก หมายเหตุ • รหัสข้อผิดพลาดที่ได้รับจะแสดงเป็น “เลขฐานสิบ [เลขฐานสิบหก]” • ตำแหน่ง IP จะแสดงเป็น “ตำแหน่ง IP (เลขฐานสิบ): ตำแหน่ง MAC (เลขฐานสิบหก)”

ตัวอย่างข้อความแสดงข้อผิดพลาด

“RHAA035: PLC1: Error has been responded for device write command (Error Code: 2 [02])”

หมายเหตุ • โปรดดูรายละเอียดเพิ่มเติมเกี่ยวกับรหัสข้อผิดพลาดที่ได้รับได้จากคู่มือของอุปกรณ์ภายนอก

ข้อความแสดงข้อผิดพลาดของไดรเวอร์นี้โดยเฉพาะมีดังนี้

เลข ID ข้อความ	ข้อความแสดงข้อผิดพลาด	สาเหตุและวิธีแก้ไข
128 RHxx128	Memory Link: Checksum does not match the data actually received (ErrorCode: 06 Destination: %s)	ผลรวมการตรวจสอบในประโยคที่กำลังส่งไม่ถูกต้อง ให้แก้ไขประโยคที่กำลังส่ง
129 RHxx129	Memory Link: Undefined code has been received (ErrorCode: 10 Destination: %s)	คำสั่งในประโยคที่กำลังส่งไม่ถูกต้อง ให้แก้ไขประโยคที่กำลังส่ง
130 RHxx130	Memory Link: The specified number of data elements does not match the number of data elements received (ErrorCode: 12 Destination: %s)	จำนวนของอีลิเมนต์ข้อมูลในประโยคที่กำลังส่งไม่ถูกต้อง ให้แก้ไขประโยคที่กำลังส่ง
146 RHxx146	Memory Link: The specified address in the system area is out of the permissible range (ErrorCode: FA Destination: %s)	การระบุพื้นที่ระบบในประโยคที่กำลังส่งไม่ถูกต้อง ให้แก้ไขประโยคที่กำลังส่ง
147 RHxx147	Memory Link: An attempt has been made to write to or read from outside the system area (ErrorCode: FB Destination: %s)	ดำเนินการนอกช่วงของพื้นที่ระบบ ให้แก้ไขประโยคที่กำลังส่ง
148 RHxx148	Memory Link:A data block of an improper format has been received by the AGP (ErrorCode: FC Destination: %s)	รูปแบบในประโยคที่กำลังส่งไม่ถูกต้อง ให้แก้ไขประโยคที่กำลังส่ง
149 RHxx149	Memory Link: The AGP has been unable to send data (ErrorCode: FF Destination: COM port name)	GP ส่งคำสั่งไม่ได้ ให้ตรวจสอบการต่อสายเคเบิล

เลข ID ข้อความ	ข้อความแสดงข้อผิดพลาด	สาเหตุและวิธีแก้ไข
150 RHxx150	Memory Link: The specified attribute code is out of the permissible range (ErrorCode: 24 Destination: %s)	การกำหนดลักษณะเฉพาะในประโยคที่กำลังส่งไม่ถูกต้อง ให้แก้ไขประโยคที่กำลังส่ง
151 RHxx151	Memory Link: The contrast cannot be adjusted with this model (ErrorCode: 25 Destination: %s)	คำสั่งการปรับคอนทราสต์ถูกส่งไปที่ GP รุ่นที่ไม่สามารถปรับคอนทราสต์ได้ โปรดตรวจสอบรุ่นของ GP
152 RHxx152	Memory Link: The specified contrast setting is out of the permissible range (ErrorCode: 26 Destination: %s)	ค่าคอนทราสต์ที่กำหนดไว้ในประโยคที่กำลังส่งไม่ถูกต้อง ให้แก้ไขประโยคที่กำลังส่ง
153 RHxx153	Memory Link: The brightness cannot be adjusted with this model (ErrorCode: 27 Destination: %s)	คำสั่งการปรับความสว่างถูกส่งไปที่ GP รุ่นที่ไม่สามารถปรับความสว่างได้ โปรดตรวจสอบรุ่นของ GP
154 RHxx154	Memory Link: The specified brightness setting is out of the permissible range (ErrorCode: 28 Destination: %s)	ค่าความสว่างที่กำหนดไว้ในประโยคที่กำลังส่งไม่ถูกต้อง ให้แก้ไขประโยคที่กำลังส่ง

*%s ที่อยู่ต่อท้าย Destination จะแสดงด้วยชื่อพอร์ต COM ของ SIO, หมายเลขพอร์ตต้นทางของ TCP และตำแหน่ง IP ปลายทางของ UDP

บันทึก

11 API การเชื่อมต่อผ่านหน่วยความจำ (การสื่อสารแบบ Ethernet)

API การเชื่อมต่อผ่านหน่วยความจำ คือ API ชนิด 32 บิตสำหรับ Windows ที่ช่วยให้คุณสามารถเข้าใช้ GP จากแอ็พพลิเคชันได้อย่างง่ายดายด้วยการใช้โปรโตคอลการเชื่อมต่อผ่านหน่วยความจำโดยที่คุณไม่จำเป็นต้องเข้าใจในรายละเอียดของการเชื่อมต่อผ่านหน่วยความจำแต่อย่างใด

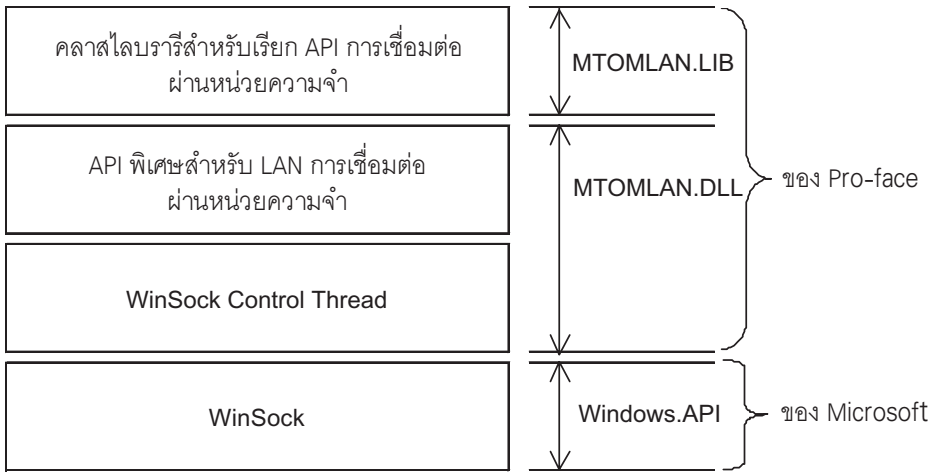
11.1 วิธีใช้ API การเชื่อมต่อผ่านหน่วยความจำ

ก่อนอื่น ผู้ใช้ API การเชื่อมต่อผ่านหน่วยความจำจะต้องสร้างช่องทางการสื่อสารไปยัง GP เสียก่อน (การสร้างช่องทางการสื่อสารเรียกว่า “การเปิดการเชื่อมต่อ”)
หลังจากทำการสื่อสารกับ GP ที่จำเป็นเสร็จสมบูรณ์แล้ว ให้ปิดช่องทางการสื่อสาร (การเชื่อมต่อ) ดังกล่าว หากคุณไม่ต้องการใช้ซ็อกเก็ตเดียวกันในการสื่อสารกับ GP ครั้งถัดไป ให้เลิกใช้ซ็อกเก็ตนั้น เมื่อต้องการสื่อสารกับ GP ครั้งถัดไป ให้เปิดการเชื่อมต่ออีกครั้ง (คุณสามารถใช้ซ็อกเก็ตซ้ำได้)

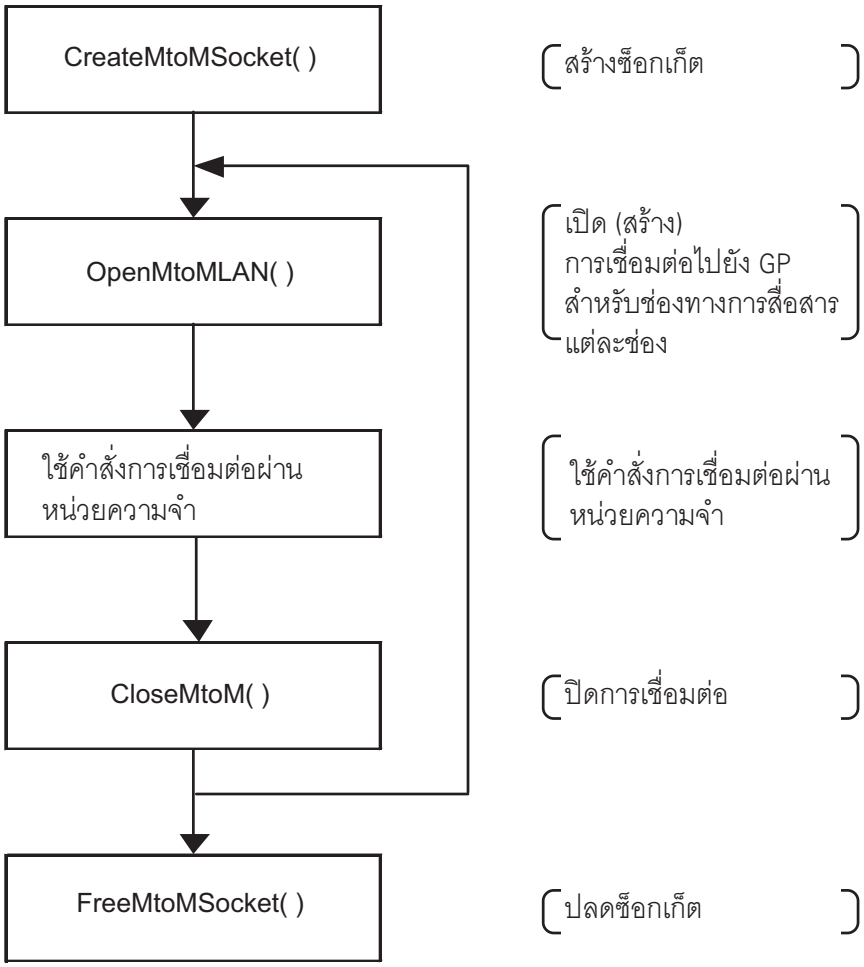
■ สภาพแวดล้อมในการพัฒนา

คอมไพเลอร์ : Microsoft Visual C++ Ver 6.0
ระบบปฏิบัติการ : Microsoft Windows 98 หรือรุ่นที่สูงกว่า
อื่น ๆ : ไฟล์ต่อไปนี้มีอยู่ในแผ่น CD-ROM ของ GP-Pro EX
ในการดูไฟล์เหล่านี้ ให้เปิดโฟลเดอร์ [MTOMLAN] ใน CD-ROM ดังกล่าวแล้วดับเบิลคลิกไฟล์ [MTOMLAN.ZIP]
MTOMAPI.H
MTOMLAN.LIB
MTOMLAN.DLL

■ ผังโครงสร้างซอฟต์แวร์ API การเชื่อมต่อผ่านหน่วยความจำ



■ **การทำงานทั่วไปของ API การเชื่อมต่อผ่านหน่วยความจำ**



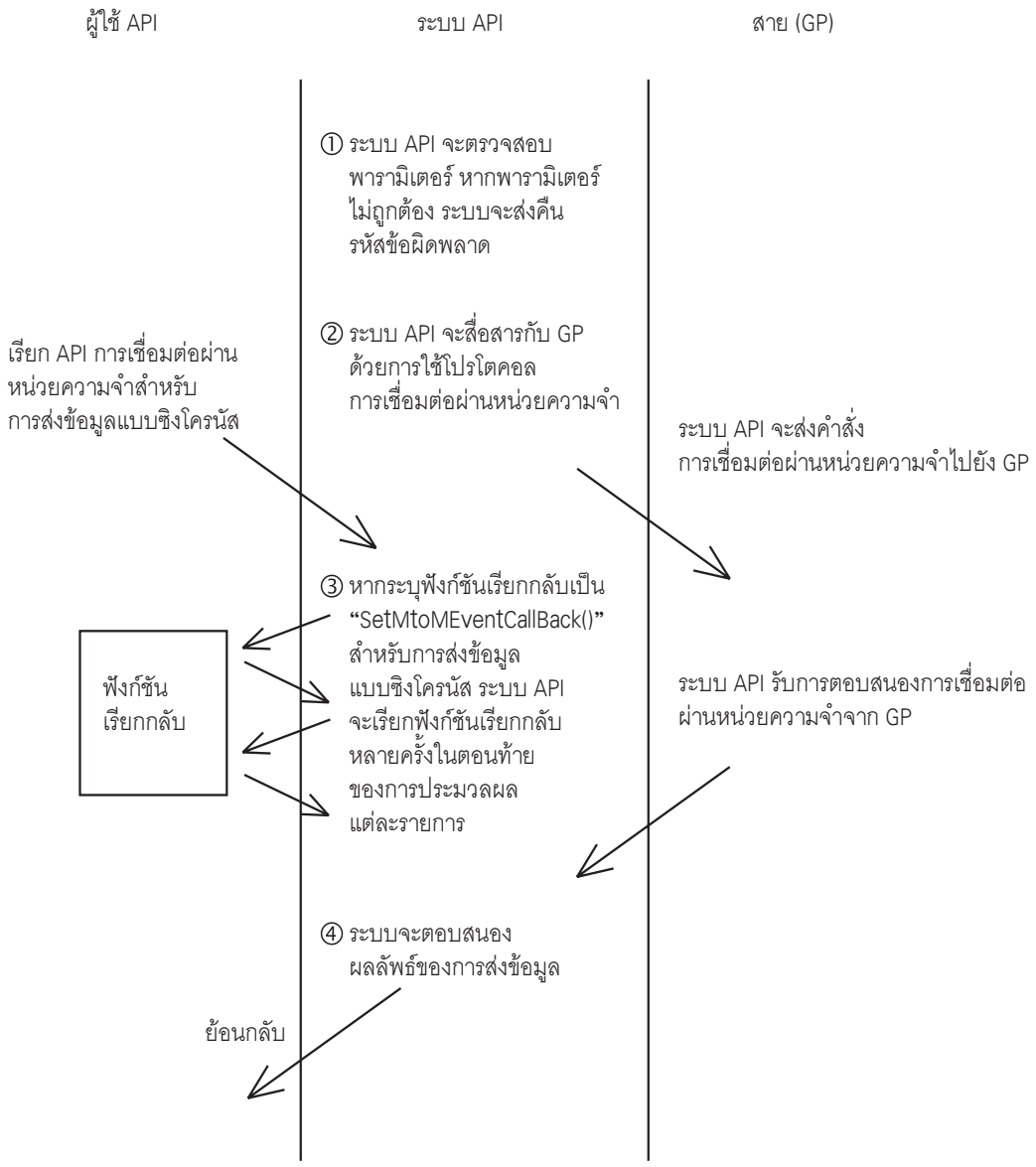
11.1.1 การส่งข้อมูลแบบซิงโครนัสและอะซิงโครนัส

“การส่งข้อมูลแบบซิงโครนัส” คือวิธีการส่งข้อมูลที่ระบบจะไม่กลับมาทำงานต่อจนกว่าการประมวลผลของ API จะสิ้นสุดลงอย่างปกติหรือผิดปกติหลังมีการเรียกคำสั่ง API
“การส่งข้อมูลแบบอะซิงโครนัส” คือวิธีการส่งข้อมูลที่ระบบจะกลับมาทำงานและพร้อมที่จะประมวลผลต่อก่อนที่การประมวลผลปัจจุบันของ API จะเสร็จสมบูรณ์
API การเชื่อมต่อผ่านหน่วยความจำรองรับได้ทั้งวิธีการส่งข้อมูลแบบซิงโครนัสและอะซิงโครนัส
พารามิเตอร์ที่สองจะระบุวิธีการส่งข้อมูลที่ใช้ระหว่างแบบซิงโครนัสหรืออะซิงโครนัส

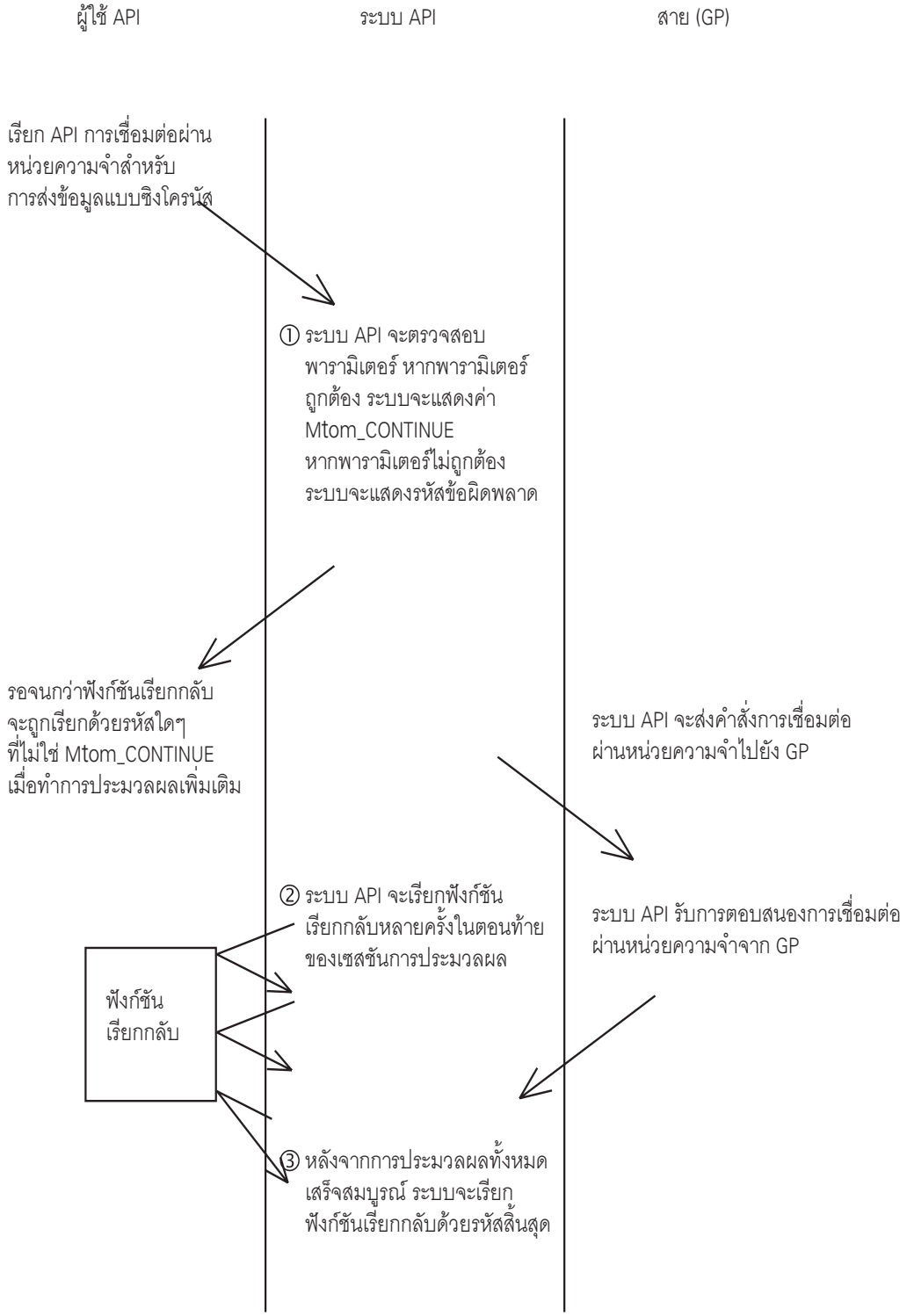
หมายเหตุ

- เมื่อพารามิเตอร์ที่สองเป็นค่าใด ๆ ที่ไม่ใช่ “MTOMCALLBACK” ระบบ API จะถูกตั้งค่าไว้ที่โหมดการส่งข้อมูลแบบซิงโครนัสโดยอัตโนมัติ
- เมื่อระบุอาร์กิวเมนต์ชนิด MTOMCALLBACK ของพารามิเตอร์ที่สองเป็น “NULL” ระบบ API จะถูกตั้งค่าไว้ที่โหมดการส่งข้อมูลแบบซิงโครนัส
- เมื่อระบุระบุอาร์กิวเมนต์ชนิด MTOMCALLBACK ของพารามิเตอร์ที่สองเป็นค่าอื่น ๆ ที่ไม่ใช่ ”NULL” ระบบ API จะถูกตั้งค่าไว้ที่โหมดการส่งข้อมูลแบบอะซิงโครนัส และจะถือว่าเป็นฟังก์ชันเรียกกลับสำหรับทำการประมวลผล

■ ขั้นตอนการส่งข้อมูลแบบซิงโครนัส



◆ **ขั้นตอนการส่งข้อมูลแบบอะซิงโครนัส**



■ การยกเลิกการส่งข้อมูลแบบอะซิงโครนัส

ในการยกเลิกการประมวลผลของ API ในระหว่างการส่งข้อมูลแบบอะซิงโครนัส ให้ดำเนินการตามวิธี 2 วิธี ดังนี้:

◆ ส่งคืน "FALSE"

API การเชื่อมต่อผ่านหน่วยความจำจะเรียกฟังก์ชันเรียกกลับในตอนท้ายของเซสชันการประมวลผลปัจจุบัน หากฟังก์ชันเรียกกลับส่งคืนค่า FALSE ในสถานะนี้ API การเชื่อมต่อผ่านหน่วยความจำจะยกเลิกการประมวลผลถัดไปได้อย่างปลอดภัย

◆ เรียก "CancelMtoM()"

หลังจากยกเลิกการประมวลผลถัดไปแล้ว API การเชื่อมต่อผ่านหน่วยความจำจะเรียกฟังก์ชันเรียกกลับด้วยรหัส "MtoM_CANCEL" ในสถานะนี้ ซ็อกเก็ตจะยังไม่มีความเสถียร ผู้ใช้ API จะต้องเรียกฟังก์ชัน FreeMtoMSocket() เพื่อปลดซ็อกเก็ตดังกล่าว หากต้องการสื่อสารต่อ ให้ใช้อีกซ็อกเก็ตหนึ่ง
ขั้นตอนนี้ใช้สำหรับบังคับแอ็พพลิเคชันการสื่อสารให้สิ้นสุดการทำงาน

■ ฟังก์ชันเรียกกลับสำหรับการส่งข้อมูลแบบอะซิงโครนัส

ในการส่งข้อมูลแบบอะซิงโครนัส ผู้ใช้ API จะต้องจัดเตรียมฟังก์ชันเรียกกลับเพื่อที่จะได้ทราบว่ากระบวนการส่งข้อมูลแบบอะซิงโครนัสได้เสร็จสมบูรณ์แล้ว
ชนิดของฟังก์ชันเรียกกลับแสดงอยู่ด้านล่างนี้

◆ ไวยากรณ์

MTOMCALLBACK FinisheMtoM(LPMtoMSOCK pMSock,int iMtoMCode)

◆ อาร์กิวเมนต์

LPMtoMSOCK pMSockใช้ซ็อกเก็ตแฮนเดิลสำหรับการประมวลผล

int iMtoMCode	ผลการประมวลผล MTOM_OK : การประมวลผลเสร็จสมบูรณ์ตามปกติ MTOM_CONTINUE : อยู่ระหว่างการประมวลผล อื่น ๆ : การประมวลผลถูกยกเลิกเนื่องจากข้อผิดพลาด

หมายเหตุ
- API ระบุรหัส MTOM_CONTINUE สำหรับพารามิเตอร์ iMtoMcode ในตอนท้ายของการประมวลผลปัจจุบัน และเรียกฟังก์ชันเรียกกลับ

11.1.2 ซ็อกเก็ตเมมเบอร์ dwUser1 และ dwUser2

ระบบ API จะไม่เขียน “dwUser1” หรือ “dwUser2” ใหม่ แต่ผู้ใช้ API สามารถใช้ซ็อกเก็ตเมมเบอร์ทั้งคู่ได้ตามต้องการ
โดยปกติ จะต้องใช้ตัวระบุสำหรับแต่ละซ็อกเก็ต

ตัวอย่าง
หากคุณออกแบบคลาส “C++” ที่รองรับซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ คุณสามารถใช้คลาสนี้กับฟังก์ชันเรียกกลับเมื่อผู้ใช้ API เรียกฟังก์ชัน CreateMtoMSocket() ด้วยบทบาทผู้สร้างของคลาสนี้ หรือสร้างซ็อกเก็ตและกำหนดตัวชี้ของคลาสนี้ของซ็อกเก็ตเป็น “dwUser1”

ตัวอย่างการทำงาน
(1) ตั้งค่าตัวชี้ “นี้” ของคลาสให้ชี้ไปที่ “dwUser1” ด้วยบทบาทผู้สร้างของคลาสนี้

(2) ลงทะเบียนฟังก์ชัน (ฟังก์ชันส่วนกลางและฟังก์ชันสถิต) ที่จะถูกเรียกกลับเป็นลำดับแรกเมื่อระบบ API แจ้งให้ทราบถึงเหตุการณ์ใด ๆ ด้วยการใช้ฟังก์ชัน SetMtoMEvent CallBack()

(3) หากมีเหตุการณ์ใด ๆ เกิดขึ้น ฟังก์ชันที่ลงทะเบียนด้วยฟังก์ชัน SetMtoMEventCallBack() (ในตัวอย่างนี้คือ EventFUncJump()) จะถูกเรียกกลับ

(4) ตัวชี้คลาสจะถูกดึงข้อมูลมาจาก “dwUser 1” ของฟังก์ชัน EventFuncJump() เสมือนว่าระบบ API ได้เรียกกลับฟังก์ชัน OnEventFunc()

(5) โดยปกติ คุณสามารถประกาศฟังก์ชัน OnEventFunc() ให้เป็นฟังก์ชันเสมือนและลบฟังก์ชันนั้นออกได้เพื่อความสะดวกในการใช้งาน

```
class CMtoMSock {
public:
LPMtoMSOCK m_pMSock ;

CMtoMSock();
~CMtoMSock();
//If you need event information from the API system, override this member.
virtual void OnEventFunc(int iCode,DWORD dwParam1,DWORD dwParam2){};// (5)
} ;

//Function to be called back when an event occurs
(3)
void CALLBACK EventFuncJump
(LPMtoMSOCK pMSock,int iCode,DWORD dwParam1,DWORD dwpara)
{
  CMSock* pCMSock ;

  pCMSock = (CMSock*)pMSock->swUser1 ;

  pCMSock->OnEventFunc(iCode,dwParam1,dwParam2) ;//}(4)
}
```

```
CMSock::CMSock(DWORD dwProtocolType)
{
  if( m_pMSock = ::CreateMtoMSocket(dwProtocolType) ){
       m_pMSock->dwUser1 = (DWORD )this ; // (1)
       ::SetMtoMEventCallBack(m_pMSock,EventFuncJump) ;// (2)
  }
 }
```

11.1.3 วิธีการส่งข้อมูล (ชนิดรายการ)

โปรโตคอล Ethernet นี้รองรับวิธีการส่งข้อมูล (ชนิดรายการ) 4 วิธี ดังต่อไปนี้:

■ การส่งข้อมูลแบบ 1:1

ระบบ API จะสื่อสารกับ GP เพียง 1 ยูนิต จึงทำให้วางใจในการสื่อสารได้ วิธีนี้ใช้โปรโตคอล TCP/IP ภายใน

ขั้นตอนพื้นฐานของการใช้รายการชนิดนี้มีดังต่อไปนี้:

(1) สร้างซ็อกเก็ตขึ้นมาหนึ่งซ็อกเก็ตโดยใช้ฟังก์ชัน CreateMtoMSocket()
(เมื่อมีการสร้างซ็อกเก็ต ระบบจะเลือกชนิดรายการนี้เป็นค่าดีฟอลต์)

(2) เปิดการเชื่อมต่อโดยใช้ฟังก์ชัน OpenMtoMLAN()

(3) ส่งข้อมูลโดยใช้ฟังก์ชัน MtoMESC_*()

(4) ปิดการเชื่อมต่อโดยใช้ฟังก์ชัน CloseMToMLAN()

(5) ปลดซ็อกเก็ตโดยใช้ฟังก์ชัน FreeMtoMSocket()

■ การส่งข้อมูลไปยังโหนดที่ไม่ระบุหมายเลข

ระบบ API จะสื่อสารกับโหนดที่ไม่ระบุหมายเลขโดยไม่ตรวจสอบว่ามีการตอบสนองหรือไม่ ดังนั้น จึงไม่สามารถวางใจในผลลัพธ์ของการสื่อสารได้ เนื่องจากวิธีการส่งข้อมูลนี้จะไม่คำนึงถึงความเร็วในการประมวลผลของโหนดปลายทาง ดังนั้นในระหว่างที่ส่งข้อมูลแบบต่อเนื่อง ข้อมูลที่ส่งจึงอาจเกิดโอเวอร์โฟลว์ได้

วิธีนี้ใช้โปรโตคอลเผยแพร่ข้อมูล UDP/IP Net ID (dwNetID) ของการเผยแพร่ตามที่ต้องการซึ่งระบุในพื้นที่ข้อมูลเน็ตเวิร์ก จะถูกนำมาใช้เป็น Net ID ของปลายทางการเผยแพร่

ขั้นตอนพื้นฐานของการใช้รายการชนิดนี้มีดังต่อไปนี้:

(1) สร้างซ็อกเก็ตขึ้นมาหนึ่งซ็อกเก็ตโดยใช้ฟังก์ชัน CreateMtoMSocket()

(2) ตั้งค่าชนิดรายการของฟังก์ชัน SetTransactionType() เป็น “การส่งข้อมูลไปยังโหนดที่ไม่ระบุหมายเลข” (B_dwTransactionType_BroadCast)

(3) เปิดการเชื่อมต่อโดยใช้ฟังก์ชัน OpenMtoMLAN() ระบุตำแหน่ง IP ของโหนดปลายทางเป็น NULL

(4) ส่งข้อมูลโดยใช้ฟังก์ชัน MtoMESC_*()

(5) ปิดการเชื่อมต่อโดยใช้ฟังก์ชัน CloseMToMLAN()

(6) ปลดซ็อกเก็ตโดยใช้ฟังก์ชัน FreeMtoMSocket()

■ การส่งข้อมูลไปยังโหนดที่ระบุ

ระบบ API จะสื่อสารกับโหนดที่ระบุ (ที่ได้รับเลือกให้เป็นเป้าหมายการประมวลผลในพื้นที่ข้อมูลเน็ตเวิร์ก)

เฉพาะการตอบสนองจากโหนดที่ได้รับการระบุให้เป็นเป้าหมายการประมวลผลที่อยู่ส่วนหัวของพื้นที่ข้อมูลเน็ตเวิร์กเท่านั้นจึงจะถือว่ามีผลใช้งาน หรืออีกนัยหนึ่งคือโหนดแรกจะถูกใช้เป็นตัวแทนของโหนดทั้งหมดในเน็ตเวิร์ก วิธีการส่งข้อมูลนี้ใช้สำหรับการส่งคำสั่งแสดงผล ไปยังโหนดต่าง ๆ หลายโหนด

หากระบุโหนดในพื้นที่ข้อมูลเน็ตเวิร์กเพียงโหนดเดียว จะใช้โปรโตคอล UDP/IP ธรรมดา (ไม่ใช่โปรโตคอลเผยแพร่ข้อมูล) หากระบุโหนดหลายโหนด จะใช้โปรโตคอลเผยแพร่ข้อมูล UDP/IP

ขั้นตอนพื้นฐานของการใช้รายการชนิดนี้มีดังต่อไปนี้:

(1) สร้างซ็อกเก็ตขึ้นมาหนึ่งซ็อกเก็ตโดยใช้ฟังก์ชัน CreateMtoMSocket()

(2) ตั้งค่าชนิดรายการของฟังก์ชัน SetTransactionType() เป็น “การส่งข้อมูลไปยังโหนดที่ไม่ระบุหมายเลข” (B_dwTransactionType_Specifict)

(3) ระบุ Net ID ของเน็ตเวิร์กเป้าหมายให้เป็น Net ID (dwNetID) ของเป้าหมายการเผยแพร่ในพื้นที่ข้อมูลเน็ตเวิร์กการเผยแพร่ (pGPNetWORkData)

(4) ระบุโหนดปลายทางในพื้นที่ข้อมูลเน็ตเวิร์ก
หากทราบโหนดปลายทางอย่างแน่ชัด ให้เรียกฟังก์ชัน MtoM_ResizeGPNetWORkData() เพื่อเปลี่ยนขนาดข้อมูลเน็ตเวิร์กและระบุตำแหน่ง IP ของโหนดปลายทาง และป้อนพารามิเตอร์ dwnodeStatus ของบันทึกโหนดของพื้นที่ข้อมูลเน็ตเวิร์กเป็น B_dwNodeStatus_Find เพื่อให้บันทึกโหนดดังกล่าวมีผลใช้งาน
หากไม่ทราบโหนดปลายทางที่แน่ชัด ให้เรียกฟังก์ชัน MtoMFS_FindNode เพื่อค้นหาโหนดต่าง ๆ ที่มีอยู่ในเน็ตเวิร์กโดยอัตโนมัติ ผลการค้นหาจะถูกเพิ่มลงในข้อมูลเน็ตเวิร์ก

หมายเหตุ
- ระหว่างที่อยู่ในโหมดการส่งข้อมูลนี้ บันทึกโหนดที่ระบุไว้ที่ส่วนหัวของพื้นที่ข้อมูลเน็ตเวิร์กจะแสดงถึงโหนดที่เป็นตัวแทนของโหนดทั้งหมดในเน็ตเวิร์กนี้

(5) ส่งข้อมูลโดยใช้ฟังก์ชัน MtoMESC_*()

(6) ปิดการเชื่อมต่อโดยใช้ฟังก์ชัน CloseMToMLAN()

(7) ปลดซ็อกเก็ตโดยใช้ฟังก์ชัน FreeMtoMSocket()

■ การส่งข้อมูลไปยังโหนดที่ระบุ (ตรวจสอบสถานะการประมวลผลของแต่ละโหนด)

ระบบ API จะสื่อสารกับโหนดที่ระบุ (ที่ได้รับเลือกให้เป็นเป้าหมายการประมวลผลในพื้นที่ข้อมูลเน็ตเวิร์ก)

หลังจากที่คำสั่งประมวลผลถูกส่งออกไป เฉพาะการตอบสนองจากโหนดที่ถูกระบุให้เป็นเป้าหมายการประมวลผลที่ส่วนหัวของพื้นที่ข้อมูลเน็ตเวิร์กเท่านั้น จึงจะถือว่ามีผลใช้งาน วิธีการส่งข้อมูลนี้แตกต่างจากวิธีที่อธิบายไว้ในหน้าก่อนนี้ เนื่องจากมีการตรวจสอบสถานะการประมวลผลของแต่ละโหนด

หากระบุโหนดในพื้นที่ข้อมูลเน็ตเวิร์กเพียงโหนดเดียว จะใช้โปรโตคอล UDP/IP ธรรมดา (ไม่ใช่โปรโตคอลเผยแพร่ข้อมูล) หากระบุโหนดหลายโหนด จะใช้โปรโตคอลเผยแพร่ข้อมูล UDP/IP

(1) สร้างซ็อกเก็ตขึ้นมาหนึ่งซ็อกเก็ตโดยใช้ฟังก์ชัน CreateMtoMSocket()

(2) ตั้งค่าชนิดรายการของฟังก์ชัน SetTransactionType() เป็น “การส่งข้อมูลไปยังโหนดที่ไม่ระบุหมายเลข” (B_dwTransactionType_Specifict)

(3) เปิดการเชื่อมต่อโดยใช้ฟังก์ชัน OpenMtoMLAN() ระบุตำแหน่ง IP ของโหนดปลายทางเป็น NULL

(4) ระบุโหนดปลายทางในพื้นที่ข้อมูลเน็ตเวิร์ก
หากทราบโหนดปลายทางอย่างแน่ชัด ให้เรียกฟังก์ชัน MtoM_ResizeGPNetWORkData() เพื่อเปลี่ยนขนาดข้อมูลเน็ตเวิร์กและระบุตำแหน่ง IP ของโหนดปลายทาง และป้อนพารามิเตอร์ dwnodeStatus ของบันทึกโหนดของพื้นที่ข้อมูลเน็ตเวิร์กเป็น B_dwNodeStatus_Find เพื่อให้บันทึกโหนดดังกล่าวมีผลใช้งาน
หากไม่ทราบโหนดปลายทางที่แน่ชัด ให้เรียกฟังก์ชัน MtoMFS_FindNode เพื่อค้นหาโหนดต่าง ๆ ที่มีอยู่ในเน็ตเวิร์กโดยอัตโนมัติ ผลการค้นหาจะถูกเพิ่มลงในพื้นที่ข้อมูลเน็ตเวิร์ก

หมายเหตุ
- ระหว่างที่อยู่ในโหมดการส่งข้อมูลนี้ บันทึกโหนดที่ระบุไว้ที่ส่วนหัวของพื้นที่ข้อมูลเน็ตเวิร์กจะแสดงถึงโหนดที่เป็นตัวแทนของโหนดทั้งหมดในเน็ตเวิร์กนี้

(5) ระบุค่าพารามิเตอร์ (dwCheckButtom) เป็น TRUE เพื่อให้โหนดนี้เป็นเป้าหมายการประมวลผลในบันทึกโหนดของพื้นที่ข้อมูลเน็ตเวิร์ก

(6) ส่งข้อมูลโดยใช้ฟังก์ชัน MtoMESC_*()

(7) ตรวจสอบสถานะโหนดของแต่ละบันทึกโหนด เพื่อตรวจสอบว่าการประมวลผลของแต่ละโหนดเสร็จสมบูรณ์ตามปกติ เมื่อระบุสถานะโหนดเป็น B_dwNodeStatus_Nothing คุณไม่ต้องสนใจบันทึกโหนดนี้ก็ได้เนื่องจากเป็นโหนดว่างเปล่า

B_dwNodeStatus_Find	:ประมวลผลเสร็จสมบูรณ์ตามปกติ
B_dwNodeStatus_NotFind	:ประมวลผลเสร็จสมบูรณ์แบบไม่ปกติ
B_dwNodeStatus_NonAction	:โหนดนี้ไม่ใช่เป้าหมายการประมวลผล หมายความว่าในขั้นตอนที่ 5 ไม่ได้ระบุ dwCheckButtom เป็น TRUE ไว้

(8) หากคุณอยากลองใหม่อีกครั้งหลังจากตรวจสอบค่า dwNodeStatus แล้ว ให้ป้อนพารามิเตอร์ dwCheckButtom เฉพาะของโหนดที่จะลองใหม่เท่านั้นด้วยค่า TRUE สำหรับโหนดอื่น ๆ ให้ป้อนค่าพารามิเตอร์นี้เป็น FALSE และปฏิบัติขั้นตอนที่ 5 และขั้นตอนถัดไปอีกครั้ง

(9) ปิดการเชื่อมต่อโดยใช้ฟังก์ชัน CloseMToMLAN()

(10)ปลดซ็อกเก็ตโดยใช้ฟังก์ชัน FreeMtoMSocket()

11.2 คำสั่งพื้นฐาน

ในส่วนนี้จะอธิบายถึงคำสั่งพื้นฐานที่ใช้สำหรับระบบ API การเชื่อมต่อผ่านหน่วยความจำ

■ รายการคำสั่งพื้นฐาน

คำสั่ง	การดำเนินการ
CreateMtoMSocket	สร้างซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำของชนิดโปรโตคอลที่ระบุ
OpenMtoMLAN	เปิดการเชื่อมต่อกับโหนดที่ระบุใน LAN การเชื่อมต่อผ่านหน่วยความจำ
CloseMtoM	ปิดการเชื่อมต่อกับโหนดปลายทาง
FreeMtoMSocket	ปลดซ็อกเก็ต
SetMtoMEventCallBack	ลงทะเบียนฟังก์ชันให้ยอมรับเหตุการณ์ เมื่อมีเหตุการณ์ใด ๆ เกิดขึ้นในระบบ API การเชื่อมต่อผ่านหน่วยความจำ
CancelMtoM	ยกเลิกการส่งข้อมูลแบบอะซิงโครนัสที่ประมวลผลในปัจจุบัน
MtoM_ResizeGPNetWorkData	เปลี่ยนแปลงขนาดของข้อมูลเน็ตเวิร์กของซ็อกเก็ต
SetTransitionType	ระบุวิธีการส่งข้อมูล (ชนิดรายการ)
GetTransitionType	เรียกวิธีการส่งข้อมูลที่ระบุในปัจจุบัน (ชนิดรายการ)
MtoMGetLastError	เรียกข้อมูลรายละเอียดข้อผิดพลาด เมื่อมีข้อผิดพลาดเกิดขึ้น

11.2.1 การสร้างซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำของโปรโตคอลที่ระบุ

ในการสร้างซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำของชนิดโปรโตคอลที่ระบุ ให้ใช้คำสั่งดังต่อไปนี้:
ระบบ API นี้จะช่วยรักษาความปลอดภัยให้แหล่งข้อมูลของซ็อกเก็ต

■ ไวยากรณ์

LPMtoMSOCK WINAPI CreateMtoMSocket(DWORD dwProtocolType)

■ ค่าที่ส่งคืน

อื่น ๆ :แฮนเดิลของซ็อกเก็ตที่สร้างขึ้น
NULL :สร้างซ็อกเก็ตไม่สำเร็จ

■ อาร์กิวเมนต์

DWORD dwProtocolType ชนิดโปรโตคอลที่จะนำมาใช้
B_ProtocolType_SIO :SIO การเชื่อมต่อผ่านหน่วยความจำ
B_ProtocolType_LAN :LAN การเชื่อมต่อผ่านหน่วยความจำ

11.2.2 การเปิดการเชื่อมต่อกับโหนดที่ระบุใน LAN การเชื่อมต่อผ่านหน่วยความจำ

ในการเปิดการเชื่อมต่อกับโหนดที่ระบุใน LAN การเชื่อมต่อผ่านหน่วยความจำ ให้ใช้คำสั่งดังต่อไปนี้:

■ ไวยากรณ์

int WINAPI OpenMtoMLAN(LPMtoMSOCK pMSock,MTOMCALLBACK pfFinish, LPCSTR szIPAddress)

■ ค่าที่ส่งคืน

เมื่อ pfFinish เป็น NULL
00 :สิ้นสุดตามปกติ
อื่นๆ :รหัสข้อผิดพลาด
เมื่อ pfFinish มีค่าใดๆ ที่ไม่ใช่ NULL
MTOM_CONTINUE :ระบบจะได้รับแจ้งถึงการร้องขอประมวลผลตามปกติ และจะได้รับแจ้งว่า การประมวลผลเสร็จสมบูรณ์เมื่อ pfFinish ถูกเรียกกลับ
อื่นๆ :รหัสข้อผิดพลาด

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ	
MTOMCALLBACK pfFinish	NULL :	ระบบ API นี้จะยังไม่เสร็จสมบูรณ์จนกว่าการประมวลผลที่ระบุจะดำเนินการเสร็จสมบูรณ์เสียก่อน (การส่งข้อมูลแบบซิงโครนัส) หลังจากการประมวลผลที่ระบุดำเนินการเสร็จสมบูรณ์แล้ว ระบบ API จะแสดงรหัสผลการประมวลผล
	ค่าอื่นๆ ที่ไม่ใช่ NULL :	ตัวชี้ไปยังฟังก์ชันจะถูกเรียกกลับหลังจากการประมวลผลเสร็จสมบูรณ์ เมื่อระบุพารามิเตอร์นี้ ระบบ API จะแสดงค่า MTOM_CONTINUE ทันทีหลังจากมีการร้องขอการประมวลผล (การส่งข้อมูลแบบอะซิงโครนัส) หลังจากที่ระบบประมวลผลเสร็จสมบูรณ์ ฟังก์ชันเรียกกลับที่ระบุไว้จะถูกเรียกด้วยซ็อกเก็ตแฮนเดิลและรหัสผลการประมวลผลที่เกี่ยวข้อง
LPCSTR szIPAddress	ตำแหน่ง IP ของโหนดปลายทาง (GP) สำหรับการสื่อสารแบบ 1:n ให้ระบุค่า NULL	

หมายเหตุ

- การระบุตำแหน่ง IP มีด้วยกัน 2 วิธีดังต่อไปนี้:
  การคั่นตำแหน่ง IP ด้วยจุด:
  ตัวอย่าง szipaddress="11.22.33.44"
  การระบุชื่อโหนดของตำแหน่ง IP
  ตัวอย่าง szipaddress="GP1"
  หากใช้วิธีนี้ คุณต้องจัดเตรียมไฟล์ HOSTS ที่อธิบายถึงตำแหน่ง IP ที่เกี่ยวข้องกับชื่อโหนดที่ระบุในโฟลเดอร์ Windows
  ตัวอย่าง ข้อมูลของ C:\Windows \ HOSTSC
  11.22.33.44 GP1

11.2.3 การปิดการเชื่อมต่อแบบ TCP กับโหนดปลายทาง

ในการปิดการเชื่อมต่อแบบ TCP กับโหนดปลายทาง ให้ใช้คำสั่งดังต่อไปนี้:

■ **ไวยากรณ์**

int WINAPI CloseMtoM(LPMtoMSOCK pMSock,MTOMCALLBACK pfFinish)

■ **ค่าที่ส่งคืน**

อื่น ๆ : แฮนเดิลของซ็อกเก็ตที่สร้างขึ้น
NULL : สร้างซ็อกเก็ตไม่สำเร็จ

■ **อาร์กิวเมนต์**

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ	
MTOMCALLBACK pfFinish	NULL:	ระบบ API นี้จะยังไม่เสร็จสมบูรณ์จนกว่าการประมวลผลที่ระบุจะดำเนินการเสร็จสมบูรณ์เสียก่อน (การส่งข้อมูลแบบซิงโครนัส) หลังจากการประมวลผลที่ระบุดำเนินการเสร็จสมบูรณ์แล้ว ระบบ API จะแสดงรหัสผลการประมวลผล
	ค่าอื่น ๆ ที่ไม่ใช่ NULL:	ตัวชี้ไปยังฟังก์ชันจะถูกเรียกกลับหลังจากการประมวลผลเสร็จสมบูรณ์ เมื่อระบุพารามิเตอร์นี้ ระบบ API จะแสดงค่า MTOM_CONTINUE ทันทีหลังจากมีการร้องขอการประมวลผล (การส่งข้อมูลแบบอะซิงโครนัส) หลังจากที่ระบบประมวลผลเสร็จสมบูรณ์ ฟังก์ชันเรียกกลับที่ระบุไว้จะถูกเรียกด้วยซ็อกเก็ตแฮนเดิลและรหัสผลการประมวลผลที่เกี่ยวข้อง

11.2.4 การปลดซ็อกเก็ต

ในการปลดซ็อกเก็ต ให้ใช้คำสั่งดังต่อไปนี้:

■ **ไวยากรณ์**

int WINAPI FreeMtoMSoket(LPMtoMSOCK pMSock)

■ **ค่าที่ส่งคืน**

อื่น ๆ : แฮนเดิลของซ็อกเก็ตที่สร้างขึ้น
NULL : สร้างซ็อกเก็ตไม่สำเร็จ

■ **อาร์กิวเมนต์**

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ

11.2.5 การลงทะเบียนฟังก์ชัน (เหตุการณ์ของระบบ API การเชื่อมต่อผ่านหน่วยความจำ)

ในการลงทะเบียนฟังก์ชันเพื่อให้ยอมรับเหตุการณ์ เมื่อมีเหตุการณ์ใด ๆ เกิดขึ้นในระบบ API การเชื่อมต่อผ่านหน่วยความจำ ให้ใช้คำสั่งดังต่อไปนี้:

■ ไวยากรณ์

int WINAPI SetMtoMEventCallBack(LPMtoMSOCK pMSock,MTOMEVENTBACK pfEventFunc)

■ ค่าที่ส่งคืน

อื่น ๆ : แฮนเดิลของซ็อกเก็ตที่สร้างขึ้น
NULL : สร้างซ็อกเก็ตไม่สำเร็จ

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ
MTOMEVENTBACK pfEventFunc	ฟังก์ชันที่จะถูกเรียกกลับเมื่อเกิดเหตุการณ์ขึ้น หากระบุ NULL ฟังก์ชันนั้นจะไม่ถูกเรียกกลับ

หากเกิดเหตุการณ์ใด ๆ ขึ้น ระบบ API จะเรียกกลับ pfEventFunc ที่ระบุ พร้อมด้วยซ็อกเก็ต รหัสและข้อมูลเหตุการณ์ (ข้อมูลชนิด 32 บิตสูงสุด 2 แพ็กเก็ต) ฟังก์ชันที่จะถูกเรียกกลับจะต้องมีรูปแบบดังต่อไปนี้:

ในระหว่างการส่งข้อมูลแบบซิงโครนัส ระบบจะเรียกฟังก์ชันเรียกกลับที่ลงทะเบียนไว้ในตอนท้ายของการประมวลผล
ในระหว่างการส่งข้อมูลแบบอะซิงโครนัส ระบบจะเรียกฟังก์ชันเรียกกลับที่ระบุเมื่อร้องขอการประมวลผล
แทนที่จะเรียกฟังก์ชันเรียกกลับที่ลงทะเบียนไว้ด้วยคำสั่งนี้

MTOMEVENTBACK EventFunc
(LPMtoMSOCK pMtoMSOCK,int iMtoMCode,DWORD dwParam1,DWORD dwParam2);

LPMtoMSOCK pMtoMSOCK	ซ็อกเก็ตแฮนเดิล
int iMtoMCode	รหัสเหตุการณ์
DWORD dwParam1	ข้อมูลที่ 1
DWORD dwParam2	ข้อมูลที่ 2

เหตุการณ์ต่อไปนี้สามารถถูกเรียกกลับได้:

รหัสเหตุการณ์	ข้อมูลที่ 1	ข้อมูลที่ 2	รายละเอียดของเหตุการณ์
MTOM_EVENT_TOUCH	รหัส T-Tag	ไม่มีความหมาย	หน้าจอสัมผัสถูกแตะ
MTOM_EVENT_CLOSED	ไม่มีความหมาย	ไม่มีความหมาย	ปิดการเชื่อมต่อแล้ว
MTOM_CONTINUE	ไม่มีความหมาย	ไม่มีความหมาย	การส่งข้อมูลแบบซิงโครนัส

11.2.6 การยกเลิกการส่งข้อมูลแบบอะซิงโครนัส (ปัจจุบัน)

ในการยกเลิกการส่งข้อมูลแบบอะซิงโครนัสที่กำลังประมวลผลในปัจจุบัน ให้ใช้คำสั่งดังต่อไปนี้:

■ ไวยากรณ์

int WINAPI CancelMtoM(LPMtoMSOCK pMSock)

■ ค่าที่ส่งคืน

อื่น ๆ :แฮนเดิลของซ็อกเก็ตที่สร้างขึ้น
NULL :สร้างซ็อกเก็ตไม่สำเร็จ

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ

หมายเหตุ
- หลังจากเรียกคำสั่ง API นี้แล้ว ซ็อกเก็ตจะไม่มีความเสถียร โปรดตรวจสอบให้แน่ใจว่าได้ทำการเรียกฟังก์ชัน FreeMtoMSocket() เพื่อปลดซ็อกเก็ตแล้ว

11.2.7 การเปลี่ยนขนาดข้อมูลเน็ตเวิร์ก

ในการเปลี่ยนขนาดของข้อมูลเน็ตเวิร์ก ให้ใช้คำสั่งดังต่อไปนี้:
หากบันทึกโหนดที่จะจัดการมีจำนวนน้อย คุณสามารถใช้คำสั่ง API นี้เพิ่มหรือลดจำนวนของบันทึกโหนดได้
การเรียกคำสั่ง API นี้จะเปลี่ยนแปลงค่าของ pGPNetWorkData ที่ระบุพื้นที่ข้อมูลเน็ตเวิร์ก pMSock

■ ไวยากรณ์

int WINAPI MtoM_ResizeGPNetWorkData(LPMtoMSOCK pMSock,DWORD dwNodeCounter)

■ ค่าที่ส่งคืน

0	: ขนาดข้อมูลเน็ตเวิร์กได้ถูกเปลี่ยนตามปกติ
อื่น ๆ	: ไม่สามารถเปลี่ยนขนาดข้อมูลเน็ตเวิร์กได้เนื่องจากหน่วยความจำมีความจุไม่เพียงพอ

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ
DWORD dwNodeCounter	จำนวนของบันทึกโหนดที่ต้องการ

11.2.8 การตั้งค่าวิธีการส่งข้อมูล (ชนิดรายการ)

ในการตั้งค่าวิธีการส่งข้อมูล (ชนิดรายการ) ให้ใช้วิธีการดังต่อไปนี้:

■ ไวยากรณ์

DWORD WINAPI SetTransctionType(LPMtoMSOCK pMSock,DWORD dwTranscitonType)

■ ค่าที่ส่งคืน

การตั้งค่าของชนิดรายการที่ยังไม่ถูกเปลี่ยนแปลง

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ
DWORD dwTranscitonType	ชนิดรายการที่จะถูกเปลี่ยนแปลง
B_dwTransctionType_Only1:	การส่งข้อมูลไปยังเฉพาะโหนดที่ระบุ 1 โหนดซึ่งได้เปิดการเชื่อมต่อแล้ว (การตั้งค่าดีฟอลต์) (ใช้โปรโตคอล TCP/IP)
B_dwTransctionType_BroadCast：	การส่งข้อมูลไปยังโหนดที่ไม่ระบุหมายเลข (ไม่ตรวจสอบการตอบสนอง) เนื่องจากวิธีการส่งข้อมูลนี้จะไม่คำนึงถึงความเร็วในการประมวลผลของโหนดปลายทาง ดังนั้นในระหว่างที่ส่งข้อมูลแบบต่อเนื่อง ข้อมูลที่ส่งจึงอาจเกิดโอเวอร์โฟลว์ได้ (ใช้โปรโตคอลเผยแพร่ข้อมูล UDP/IP)
B_dwTransctionType_Specific：	การส่งข้อมูลไปยังโหนดที่ระบุ (ที่ถูกเลือกเป็นเป้าหมายการประมวลผลในพื้นที่ข้อมูลเน็ตเวิร์ก) เฉพาะการตอบสนองจากโหนดที่ได้รับการระบุให้เป็นเป้าหมายการประมวลผลที่อยู่ส่วนหัวของพื้นที่ข้อมูลเน็ตเวิร์กเท่านั้นจึงจะถือว่ามีผลใช้งาน หรืออีกนัยหนึ่งคือโหนดแรกจะถูกใช้เป็นตัวแทนของโหนดทั้งหมดในเน็ตเวิร์ก วิธีการส่งข้อมูลนี้ใช้สำหรับการส่งคำสั่งแสดงผล ไปยังโหนดต่าง ๆ หลายโหนด (ใช้โปรโตคอลเผยแพร่ข้อมูล UDP/IP)
B_dwTransctionType_SpecificCheck：	การส่งข้อมูลไปยังโหนดที่ระบุ (ที่ถูกเลือกเป็นเป้าหมายการประมวลผลในพื้นที่ข้อมูลเน็ตเวิร์ก) เฉพาะการตอบสนองจากโหนดที่ได้รับการระบุให้เป็นเป้าหมายการประมวลผลที่อยู่ส่วนหัวของพื้นที่ข้อมูลเน็ตเวิร์กเท่านั้นจึงจะถือว่ามีผลใช้งาน วิธีการส่งข้อมูลนี้แตกต่างจากวิธีที่ระบุโดย B_dwTransctionType_Specific เนื่องจากมีการตรวจสอบผลการประมวลผลของแต่ละโหนดด้วย วิธีนี้ใช้สำหรับตรวจสอบผลการประมวลผลของของโหนดต่าง ๆ อย่างใกล้ชิด (เช่น ในกระบวนการถ่ายโอนไฟล์) (ใช้โปรโตคอลเผยแพร่ข้อมูล UDP/IP)

11.2.9 การเรียกข้อมูลวิธีการส่งข้อมูลปัจจุบัน

ในการเรียกข้อมูลวิธีการส่งข้อมูลที่ระบุในปัจจุบัน (ชนิดรายการ) ให้ใช้คำสั่งดังต่อไปนี้:

■ **ไวยากรณ์**

DWORD WINAPI GetTranscitonType(LPMtoMSOCK pMSock)

■ **ค่าที่ส่งคืน**

การตั้งค่าของวิธีการส่งข้อมูลที่ระบุในปัจจุบัน (ชนิดรายการ)
โปรดดูรายละเอียดในพารามิเตอร์ dwTransctionType ของฟังก์ชัน SetTransctionType()

■ **อาร์กิวเมนต์**

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ

11.2.10 การเรียกข้อมูลรายละเอียดข้อผิดพลาดปัจจุบัน

เมื่อเกิดข้อผิดพลาดขึ้น ให้ใช้คำสั่งดังต่อไปนี้เพื่อเรียกข้อมูลรายละเอียดของข้อผิดพลาด

■ ไวยากรณ์

DWORD WINAPI MtoMGetLastError(LPMtoMSOCK pMSock)

■ ค่าที่ส่งคืน

เมื่อเกิดข้อผิดพลาดขึ้นเมื่อใช้ระบบ API การเชื่อมต่อผ่านหน่วยความจำ ระบบจะแสดงรายละเอียดของข้อผิดพลาด

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ

■ คำอธิบาย

โดยทั่วไป รายละเอียดของข้อผิดพลาดจะถูกแบ่งออกเป็น 2 ชนิด คือ ข้อผิดพลาดในการตอบสนองจาก GP และข้อผิดพลาดเนื่องจากปัญหาของสายเคเบิล

หากเกิดข้อผิดพลาดชนิดแรกขึ้น ระบบจะแสดงค่า 9999 หรือน้อยกว่า ค่า “0” หมายความว่าการประมวลผลได้เสร็จสมบูรณ์ตามปกติ

หากเกิดข้อผิดพลาดชนิดหลังขึ้น ระบบจะแสดงค่า 10000 หรือมากกว่า หรือกล่าวให้ชัดเจนยิ่งขึ้นคือ เนื่องจาก API การเชื่อมต่อผ่านหน่วยความจำใช้ Winsock ของ Microsoft Visual C++ เป็นการภายใน ดังนั้นค่าที่ระบบแสดงก็คือรหัสข้อผิดพลาดนั่นเอง

◆ รหัสข้อผิดพลาดที่เกี่ยวกับ GP

Cf. คู่มืออ้างอิงสำหรับ GP-Pro EX

◆ รหัสข้อผิดพลาดที่เกี่ยวกับ Winsock

รหัส	ข้อผิดพลาด	รหัส	ข้อผิดพลาด
10004	WSAEINTR	10053	WSAECONNABORTED
10009	WSAEBADF	10054	WSAECONNRESET
10013	WSAEACCES	10055	WSAENOBUFS
10014	WSAEFAULT	10056	WSAEISCONN
10022	WSAEINVAL	10057	WSAENOTCONN
10024	WSAEMFILE	10058	WSAESHUTDOWN
10035	WSAEWOULDBLOCK	10059	WSAETOOMANYREFS
10036	WSAEINPROGRESS	10060	WSAETIMEDOUT
10037	WSAEALREADY	10061	WSAECONNREFUSED
10038	WSAENOTSOCK	10062	WSAELOOP
10039	WSAEDESTADDRREQ	10063	WSAENAMETOOLONG
10040	WSAEMSGSIZE	10064	WSAEHOSTDOWN

รหัส	ข้อผิดพลาด	รหัส	ข้อผิดพลาด
10041	WSAEPROTOTYPE	10065	WSAEHOSTUNREACH
10042	WSAENOPROTYPE	10066	WSAENOTEMPTY
10043	WSAEPROTONOSUPPORT	10067	WSAEPROCLIM
10044	WSAESOCKTNOSUPPORT	10068	WSAEUSERS
10045	WSAEOPNOTSUPP	10069	WSAEDQUOT
10046	WSAEPFNOSUPPORT	10070	WSAESTALE
10047	WSAEAFNOSUPPORT	10071	WSAEREMOTE
10048	WSAEADDRINUSE	10091	WSASYSNOTREADY
10049	WSAEADDRNOTAVAIL	10092	WSAVERNOTSUPPROTED
10050	WSAENETDOWN	10093	WSANOTINITIALISED
10051	WSAENETUNREACH	10101	WSAEDISCON
10052	WSAENETRESET	-	-

11.3 คำสั่งของโหมดการแสดงผล

ในส่วนนี้จะอธิบายถึงคำสั่งแสดงผลที่ใช้กับระบบ API การเชื่อมต่อผ่านหน่วยความจำ

■ รายการคำสั่งแสดงผล

คำสั่ง	การดำเนินการ
MtoMESC_W	เขียนข้อมูลลงในพื้นที่ระบบ
MtoMESC_R	อ่านข้อมูลจากพื้นที่ระบบ
MtoMESC_I	ตรวจสอบว่ามีการแตะหน้าจอสัมผัสหรือไม่
MtoMESC_SetContrast	ตั้งค่าความสว่าง/คอนทราสต์
MtoMESC_GetContrast	เรียกข้อมูลการตั้งค่าความสว่าง/คอนทราสต์

11.3.1 การเขียนข้อมูลลงในพื้นที่ระบบ

ในการเขียนข้อมูลลงในพื้นที่ระบบ ให้ใช้คำสั่งดังต่อไปนี้:

■ ไวยากรณ์

int WINAPI MtoMESC_W
(LPMtoMSOCK pMSock,MTOMCALLBACK pfFinish,WORD wAddress,INT iDataCount,WORD* pwData)

■ ค่าที่ส่งคืน

เมื่อ pfFinish เป็น NULL
00 : สิ้นสุดตามปกติ
อื่นๆ : รหัสข้อผิดพลาด
เมื่อ pfFinish มีค่าใดๆ ที่ไม่ใช่ NULL

MTOM_CONTINUE:	ระบบจะได้รับแจ้งถึงการร้องขอประมวลผลตามปกติ และจะได้รับแจ้งว่าการประมวลผลเสร็จสมบูรณ์เมื่อ pfFinish ถูกเรียกกลับ
อื่นๆ:	รหัสข้อผิดพลาด

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock		แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ
MTOMCALLBACK pfFinish	NULL:	ระบบ API นี้จะยังไม่เสร็จสมบูรณ์จนกว่าการประมวลผลที่ระบุจะดำเนินการเสร็จสมบูรณ์เสียก่อน (การส่งข้อมูลแบบซิงโครนัส) หลังจากการประมวลผลที่ระบุดำเนินการเสร็จสมบูรณ์แล้ว ระบบ API จะแสดงรหัสผลการประมวลผล
	ค่าอื่นๆ ที่ไม่ใช่ NULL:	ตัวชี้ไปยังฟังก์ชันจะถูกเรียกกลับหลังจากการประมวลผลเสร็จสมบูรณ์ เมื่อระบุพารามิเตอร์นี้ ระบบ API จะแสดงค่าMTOM_CONTINUE ทันทีหลังจากมีการร้องขอการประมวลผล (การส่งข้อมูลแบบอะซิงโครนัส)

หลังจากที่ระบบประมวลผลเสร็จสมบูรณ์
ฟังก์ชันเรียกกลับที่ระบุไว้จะถูกเรียกด้วยซ็อกเก็ต
แฮนเดิลและรหัสผลการประมวลผลที่เกี่ยวข้อง

WORD wAddress	ระบุตำแหน่งของพื้นที่ระบบที่จะเขียนข้อมูล 0000h ถึง 0FFFh
INT iDataCount	ระบุจำนวนแพ็กเก็ตข้อมูลที่จะถูกเขียน 0001h ถึง 0200h (1 ถึง 512)
WORD* pwData	ข้อมูลที่จะถูกเขียน

11.3.2 การอ่านข้อมูลจากพื้นที่ระบบ

ในการอ่านข้อมูลจากพื้นที่ระบบ ให้ใช้คำสั่งดังต่อไปนี้:

■ ไวยากรณ์

int WINAPI MtoMESC_R
(LPMtoMSOCK pMSock,MTOMCALLBACK pfFinish,WORD wAddress,INT iDataCount,WORD pwoData)

■ ค่าที่ส่งคืน

เมื่อ pfFinish เป็น NULL
00 : สิ้นสุดตามปกติ
อื่นๆ : รหัสข้อผิดพลาด
เมื่อ pfFinish มีค่าใดๆ ที่ไม่ใช่ NULL
MTOM_CONTINUE :ระบบจะได้รับแจ้งถึงการร้องขอประมวลผล
และจะได้รับแจ้งว่าการประมวลผลเสร็จสมบูรณ์เมื่อ pfFinish ถูกเรียกกลับ
อื่นๆ :รหัสข้อผิดพลาด

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock		แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ
MTOMCALLBACK pfFinish	NULL :	ระบบ API นี้จะยังไม่เสร็จสมบูรณ์จนกว่าการประมวลผลที่ระบุจะดำเนินการเสร็จสมบูรณ์เสียก่อน (การส่งข้อมูลแบบซิงโครนัส) หลังจากการประมวลผลที่ระบุดำเนินการเสร็จสมบูรณ์แล้ว ระบบ API จะแสดงรหัสผลการประมวลผล
	ค่าอื่นๆ ที่ไม่ใช่ NULL :	ตัวชี้ไปยังฟังก์ชันจะถูกเรียกกลับหลังจากการประมวลผลเสร็จสมบูรณ์ เมื่อระบุพารามิเตอร์นี้ ระบบ API จะแสดงค่าMTOM_CONTINUE ทันทีหลังจากมีการร้องขอการประมวลผล (การส่งข้อมูลแบบอะซิงโครนัส) หลังจากที่ระบบประมวลผลเสร็จสมบูรณ์ ฟังก์ชันเรียกกลับที่ระบุไว้จะถูกเรียกด้วยซ็อกเก็ตแฮนเดิลและรหัสผลการประมวลผลที่เกี่ยวข้อง
WORD wAddress	ระบุตำแหน่งของพื้นที่ระบบที่จะอ่านข้อมูล 0000h ถึง 0FFFh	
INT iDataCount	ระบุจำนวนแพ็กเก็ตข้อมูลที่จะถูกอ่าน 0001h ถึง 0200h (1 ถึง 512)	
WORD pwoData	ตำแหน่งที่จัดเก็บข้อมูลที่อ่านได้	

หมายเหตุ • ระบบ API นี้จะไม่ตรวจสอบขนาดบัฟเฟอร์ที่ระบุโดย pwoData ผู้ใช้ API จะต้องจัดเตรียมขนาดบัฟเฟอร์ให้เพียงพอ

11.3.3 การตรวจสอบการป้อนข้อมูลทางหน้าจอสัมผัส

ในการตรวจสอบว่ามีการแตะหน้าจอสัมผัสหรือไม่นั้น ให้ใช้คำสั่งดังต่อไปนี้:
หลังจากที่การประมวลผลของระบบ API นี้เสร็จสมบูรณ์ตามปกติแล้ว ให้ตรวจสอบค่า pbHave
หากส่วนนี้มีค่าเป็น TRUE ให้ดูที่ค่า pdwCode

■ ไวยากรณ์

int WINAPI MtoMESC_I
(LPMtoMSOCK pMSock,MTOMCALLBACK pfFinish,BOOL* pbHave,DWORD *pdwCode)

■ ค่าที่ส่งคืน

เมื่อ pfFinish เป็น NULL
00 : สิ้นสุดตามปกติ
อื่น ๆ : รหัสข้อผิดพลาด
เมื่อ pfFinish มีค่าใด ๆ ที่ไม่ใช่ NULL
MTOM_CONTINUE :ระบบจะได้รับแจ้งถึงการร้องขอประมวลผล และจะได้รับแจ้งว่า การประมวลผลเสร็จสมบูรณ์เมื่อ pfFinish ถูกเรียกกลับ
อื่น ๆ :รหัสข้อผิดพลาด

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ	
MTOMCALLBACK pfFinish	NULL:	ระบบ API นี้จะยังไม่เสร็จสมบูรณ์จนกว่าการประมวลผลที่ระบุจะดำเนินการเสร็จสมบูรณ์เสียก่อน (การส่งข้อมูลแบบซิงโครนัส) หลังจากการประมวลผลที่ระบุดำเนินการเสร็จสมบูรณ์แล้ว ระบบ API จะแสดงรหัสผลการประมวลผล
	ค่าอื่น ๆ ที่ไม่ใช่ NULL:	ตัวชี้ไปยังฟังก์ชันจะถูกเรียกกลับหลังจากการประมวลผลเสร็จสมบูรณ์ เมื่อระบุพารามิเตอร์นี้ ระบบ API จะแสดงค่าMTOM_CONTINUE ทันทีหลังจากมีการร้องขอการประมวลผล (การส่งข้อมูลแบบอะซิงโครนัส) หลังจากที่ระบบประมวลผลเสร็จสมบูรณ์ ฟังก์ชันเรียกกลับที่ระบุไว้จะถูกเรียกด้วยซ็อกเก็ตแฮนเดิลและรหัสผลการประมวลผลที่เกี่ยวข้อง
BOOL* pbHave	ส่วนนี้จะระบุว่ามีการแตะหน้าจอสัมผัสหรือไม่ หากส่วนนี้มีค่าเป็น “TRUE” หลังจากการประมวลผลของระบบ API นี้เสร็จสมบูรณ์ หมายความว่ามีการแตะที่หน้าจอสัมผัส และรหัสที่เกี่ยวข้องจะถูกป้อนเป็น pdwCode	
DWORD *pdwCode	เมื่อมีการแตะที่หน้าจอสัมผัส รหัสที่เกี่ยวข้องจะถูกป้อนในส่วนนี้	

11.3.4 การตั้งค่าความสว่าง/คอนทราสต์

ในการตั้งค่าความสว่าง/คอนทราสต์ ให้ใช้คำสั่งดังต่อไปนี้:

■ ไวยากรณ์

DWORD WINAPI SetContrase
(LPMtoMSOCK pMSock,MTOMCALLBACK pfFinish, DWORD dwContrast, DWORD dwLight)

■ ค่าที่ส่งคืน

การตั้งค่าของความสว่าง/คอนทราสต์ที่จะถูกเปลี่ยนแปลง

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ	
MTOMCALLBACK pfFinish	NULL:	ระบบ API นี้จะยังไม่เสร็จสมบูรณ์จนกว่าการประมวลผลที่ระบุจะดำเนินการเสร็จสมบูรณ์เสียก่อน (การส่งข้อมูลแบบซิงโครนัส) หลังจากการประมวลผลที่ระบุดำเนินการเสร็จสมบูรณ์แล้ว ระบบ API จะแสดงรหัสผลการประมวลผล
	ค่าอื่น ๆ ที่ไม่ใช่ NULL:	ตัวชี้ไปยังฟังก์ชันจะถูกเรียกกลับหลังจากการประมวลผลเสร็จสมบูรณ์ เมื่อระบุพารามิเตอร์นี้ ระบบ API จะแสดงค่าMTOM_CONTINUE ทันทีหลังจากมีการร้องขอการประมวลผล (การส่งข้อมูลแบบอะซิงโครนัส) หลังจากที่ระบบประมวลผลเสร็จสมบูรณ์ ฟังก์ชันเรียกกลับที่ระบุไว้จะถูกเรียกด้วยซ็อกเก็ตแฮนเดิลและรหัสผลการประมวลผลที่เกี่ยวข้อง
dwContrast	การปรับคอนทราสต์ (0000h ถึง 0007h) 0: สว่าง ถึง 7: มืด “FFFFFFFFh” หมายถึง “ไม่มีการตั้งค่า” (ไม่สามารถปรับค่าคอนทราสต์ในรุ่นนี้ได้)	
dwLight	การปรับความสว่าง (0000h ถึง 0007h) 0: สว่าง ถึง 7: มืด “FFFFFFFFh” หมายถึง “ไม่มีการตั้งค่า” (ไม่สามารถปรับค่าความสว่างในรุ่นนี้ได้)	

หมายเหตุ
- โปรดดูรายละเอียดเกี่ยวกับช่วงของความสว่าง/คอนทราสต์ได้จากตารางความสว่าง/คอนทราสต์
☞ ตารางความสว่าง/คอนทราสต์ (หน้า 45)

11.3.5 การเรียกข้อมูลความสว่าง/คอนทราสต์

การเรียกข้อมูลการตั้งค่าปัจจุบันของความสว่าง/คอนทราสต์ ให้ใช้คำสั่งดังต่อไปนี้:

■ ไวยากรณ์

DWORD WINAPI GetContrase
(LPMtoMSOCK pMSock,MTOMCALLBACK pfFinish, DWORD*dwContrast, DWORD *dwLight)

■ ค่าที่ส่งคืน

การตั้งค่าความสว่าง/คอนทราสต์ปัจจุบัน

■ อาร์กิวเมนต์

LPMtoMSOCK pMSock	แฮนเดิลของซ็อกเก็ตการเชื่อมต่อผ่านหน่วยความจำ	
MTOMCALLBACK pfFinish	NULL:	ระบบ API นี้จะยังไม่เสร็จสมบูรณ์จนกว่าการประมวลผลที่ระบุจะดำเนินการเสร็จสมบูรณ์เสียก่อน (การส่งข้อมูลแบบซิงโครนัส) หลังจากการประมวลผลที่ระบุดำเนินการเสร็จสมบูรณ์แล้ว ระบบ API จะแสดงรหัสผลการประมวลผล
	ค่าอื่น ๆ ที่ไม่ใช่ NULL:	ตัวชี้ไปยังฟังก์ชันจะถูกเรียกกลับหลังจากการประมวลผลเสร็จสมบูรณ์ เมื่อระบุพารามิเตอร์นี้ ระบบ API จะแสดงค่าMTOM_CONTINUE ทันทีหลังจากมีการร้องขอการประมวลผล (การส่งข้อมูลแบบอะซิงโครนัส) หลังจากที่ระบบประมวลผลเสร็จสมบูรณ์ ฟังก์ชันเรียกกลับที่ระบุไว้จะถูกเรียกด้วยซ็อกเก็ตแฮนเดิลและรหัสผลการประมวลผลที่เกี่ยวข้อง
dwContrast	ค่าปัจจุบันของคอนทราสต์ (0000h ถึง 0007h) 0: สว่าง ถึง 7: มืด “FFFFFFFFh” หมายถึง “ไม่มีการตั้งค่า” (ไม่สามารถปรับค่าคอนทราสต์ในรุ่นนี้ได้)	
dwLight	ค่าปัจจุบันของความสว่าง (0000h ถึง 0007h) 0: สว่าง ถึง 7: มืด “FFFFFFFFh” หมายถึง “ไม่มีการตั้งค่า” (ไม่สามารถปรับค่าความสว่างในรุ่นนี้ได้)	

หมายเหตุ
- โปรดดูรายละเอียดเกี่ยวกับช่วงของความสว่าง/คอนทราสต์ได้จากตารางความสว่าง/คอนทราสต์
☞ ตารางความสว่าง/คอนทราสต์ (หน้า 45)

11.3.6 รายการรหัสข้อผิดพลาดของค่าส่งคืนของ API

	รหัส	คำอธิบาย
MTOM_OK	00	การประมวลผลเสร็จสมบูรณ์ตามปกติ
MTOM_CONTINUE	01	อยู่ระหว่างการประมวลผล
MTOM_USERS_STOPED	03	การประมวลผลถูกยกเลิกโดยแอ็พพลิเคชันของผู้ใช้ (ฟังก์ชัน MtoMStop() ถูกเรียก แต่ฟังก์ชันเรียกกลับส่งคืนค่า FALSE)
MTOM_EVENT_TOUCH	40	มีการแตะหน้าจอสัมผัส (ไม่ใช่ค่าส่งคืนของระบบ API แต่ฟังก์ชันเรียกกลับได้รับแจ้งถึงรหัสนี้ เมื่อเหตุการณ์ที่ลงทะเบียนไว้ด้วยฟังก์ชัน SetMtoMEventCallBack()) ได้เกิดขึ้น
MTOM_EVENT_CLOSED	41	ปิดการเชื่อมต่อแล้ว
MTOM_ERROR	80	ข้อผิดพลาดในการตอบสนองจากยูนิต GP
MTOM_ERROR_INVALID	81	เกิดข้อผิดพลาดพารามิเตอร์ API ขึ้น หรือ API ถูกเรียกอย่างไม่ถูกต้อง
MTOM_ERROR_LAN	82	เกิดข้อผิดพลาดที่สายเคเบิล (Winstock ส่งคืนรหัสข้อผิดพลาด)
MTOM_ERROR_TOUT_RES	83	ข้อผิดพลาดไทม์เอาต์ของการตอบสนอง
MTOM_ERROR_TOUT_CHAR	84	ข้อผิดพลาดไทม์เอาต์ของการส่งข้อมูลอักขระต่ออักขระ (การส่งเฟรมข้อมูลจาก GP ถูกขัดจังหวะ)
MTOM_ERROR_NAK	85	GP ส่งคืน NAK

หมายเหตุ
- เมื่อระบบได้รับข้อผิดพลาดในการตอบสนองจากยูนิต GP ให้เรียกฟังก์ชัน MtoMGetLastError() เพื่อดูรายละเอียดของข้อผิดพลาด

12 โปรแกรมตัวอย่าง (การสื่อสารแบบอนุกรม)

12.1 ระบบตัวอย่าง

ในส่วนนี้จะแสดงตัวอย่างโปรแกรมของเครื่องโฮสต์และการตั้งค่าพาร์ทต่าง ๆ ของ GP ที่จำเป็นต่อการส่งข้อมูลระหว่าง GP และเครื่องโฮสต์ นอกจากนี้เมื่อพาร์ทที่ตั้งค่าที่ด้านล่างนี้ทำงานด้วยโปรแกรมตัวอย่างจะแสดงการเปลี่ยนหน้าจอของ GP ด้วย
ให้ปฏิบัติตามขั้นตอนต่อไปนี้เพื่อสร้างหน้าจอดังที่แสดงอยู่ที่ด้านล่างนี้

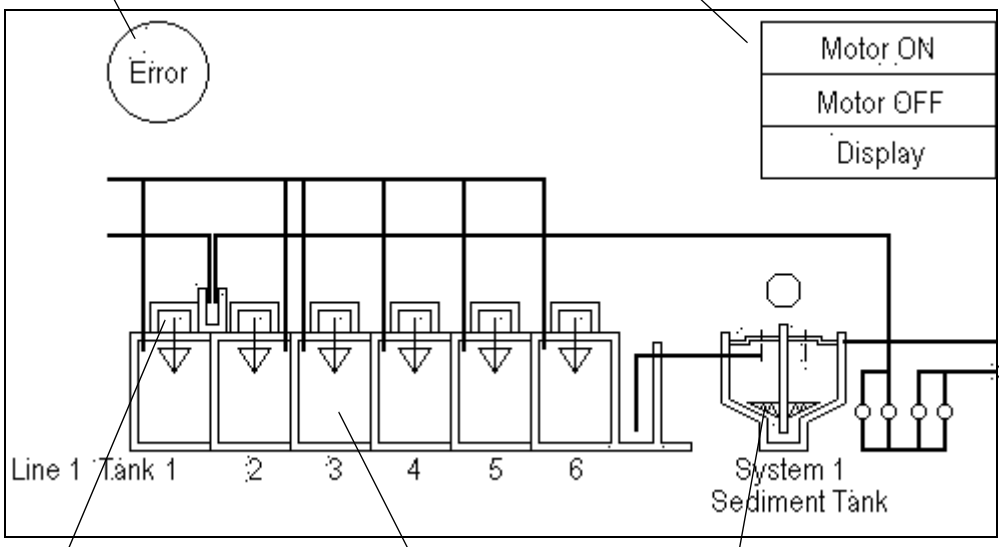
เมื่อกดสวิตช์ [Motor ON], [Motor OFF], [Display] หรือ [Error] รหัสสัญญาณอินเตอร์รัพต์ของสวิตช์ที่ถูกกดนั้นจะถูกส่งออกไปที่ระบบโฮสต์ เพื่อเริ่มการทำงานต่อไปนี้

คำอธิบายสวิตช์
[Motor ON] .....เปิดมอเตอร์เพื่อจ่ายตะกอนเข้าสู่ถังตกตะกอนเป็นปริมาณ 50%
[Motor OFF].....ปิดมอเตอร์
[Display] .....ปริมาณ 0% ของตะกอนที่จ่ายเข้าสู่ถังตกตะกอน
[Error]..... มีการจ่ายตะกอนเข้าสู่ถังตกตะกอนเป็นปริมาณเพียง 20% เท่านั้น

■ ตัวอย่างระบบ

สวิตช์ข้อผิดพลาด
เมื่อกดสวิตช์
ไฟสัญญาณจะสว่างขึ้น

สวิตช์ต่อไปนี้คือสวิตช์เปิดมอเตอร์
สวิตช์หยุดมอเตอร์ และสวิตช์แสดง
ระดับของถังตกตะกอน เมื่อกดสวิตช์
ไฟสัญญาณของสวิตช์ที่กดจะสว่างขึ้น



เมื่อกดสวิตช์ Motor ON
เครื่องหมายจะปรากฏขึ้น

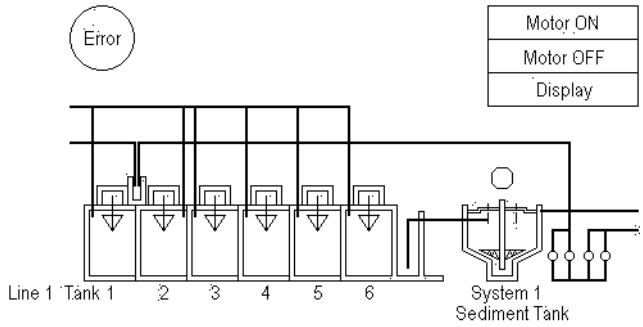
เมื่อโปรแกรมทำงาน ภาพถัง
ที่มีวัสดุอยู่ข้างในจะปรากฏขึ้น

เมื่อกดสวิตช์ [Motor ON],
[Display] และ [Error] จะแสดง
ระดับตะกอนในปัจจุบัน

■ การสร้างหน้าจอ

(1) ใช้ GP-Pro EX เพื่อสร้างหน้าจอ

หน้าจอนี้จะแสดงขึ้นขณะ GP กำลังทำงาน



(2) ใช้ GP-Pro EX เพื่อตั้งค่าพาร์ท

■ ตัวอย่างการตั้งค่าพาร์ท

รายการสวิตช์

หมายเลขหน้าจอ	ชื่อพาร์ท	สวิตช์	ตำแหน่งเวิร์ด	Word Action	หมายเลขที่กำหนดไว้คงที่
Base Screen 1	Motor ON	Word Switch	#MEMLINK 13	Write Data 16 bit Dec	0031
	Motor OFF				0032
	Display				0033
	Error				0034

◆ **แผนผังตำแหน่ง**

พาร์ทที่แสดงในตัวอย่างการตั้งค่าพาร์ทจะถูกจัดสรรไปยังตำแหน่งที่เกี่ยวข้องกับพาร์ทนั้น ๆ ดังต่อไปนี้

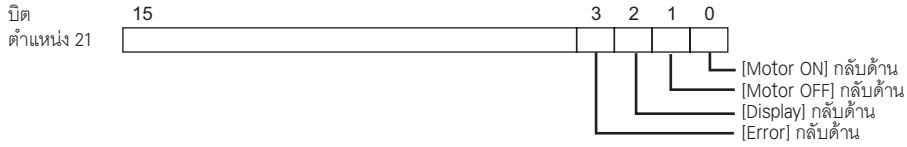
สวิตช์ -> ตำแหน่ง 13

การเขียนข้อมูลลงในตำแหน่ง 13 (อินเตอร์รัพต์) ทำให้มีการส่งออกรหัส 1 ไบต์ล่างจากพอร์ต RS232C ด้วยเหตุนี้ สวิตช์ (พาร์ท) จึงใช้การเขียนข้อมูลเวิร์ด

Motor ON.....เขียนเวิร์ด 0031ลงในตำแหน่ง 13
Motor OFF.....เขียนเวิร์ด 0032 ลงในตำแหน่ง 13
Display.....เขียนเวิร์ด 0033 ลงในตำแหน่ง 13
Error .....เขียนเวิร์ด 0034 ลงในตำแหน่ง 13

ถัง -> ตำแหน่ง 20

สวิตช์ -> ตำแหน่ง 21



มอเตอร์ -> ตำแหน่ง 22

ถังตกตะกอน -> ตำแหน่ง 23

(3) บริษัทเจ้าของเครื่องโฮสต์จะสร้างโปรแกรมถ่ายโอนข้อมูลระหว่าง GP และเครื่องโฮสต์

◆โปรแกรมตัวอย่าง

ตัวอย่าง เมื่อใช้เครื่อง IBM PC/AT-compatible และภาษา C:

```
/*****************************************************************************/
/*  */
/*  GP series     โปรแกรมตัวอย่างสำหรับการสื่อสารแบบเชื่อมต่อผ่านหน่วยความจำ*/
/*  */
/*****************************************************************************/

#include<stdio.h>
#include<dos.h>
#include<string.h>
#include<stdlib.h>
#include<conio.h>

#define  data_size_str2          20        /*ขนาดข้อมูลของ str2 คือ 20 ไบต์*/
#define  data_size_wr_data       24        /*ขนาดข้อมูลของ wr_data คือ 24 ไบต์*/

#define  serial_port_BIOS        0x14      /*BIOS พอร์ตอนุกรมของ PC*/
#define  serial_port_number      0x00      /*หมายเลขพอร์ตอนุกรมที่ใช้*/
#define  serial_port_INT         0xE7      /*พอร์ตอนุกรมเริ่มต้นการทำงาน*/
#define  serial_port_parameter   0xE7      /*9600bps,8bit,stopbit;1,parity;none*/

#define  get_status              0x03      /*ได้รับสถานะของพอร์ตอนุกรม*/
#define  serial_port_write       0x01      /*พอร์ตอนุกรมถูกเขียนข้อมูล*/
#define  serial_port_read        0x02      /*พอร์ตอนุกรมถูกอ่านข้อมูล*/

#define  status_bit_6000         0x60000   /*บิตสถานะพอร์ต บิต 13 และบิต14*/
#define  status_bit_0020         0x0020    /*บิตสถานะพอร์ต บิต 5*/

/*****************************************************************************/
/*                  การตั้งค่าการสื่อสารสำหรับ SI0*/
/*****************************************************************************/
void  open_SIO (void);/*การตั้งค่าการสื่อสารสำหรับ RS232C*/

/*****************************************************************************/
/*                การได้รับและการระบุสถานะพอร์ต*/
/*****************************************************************************/
int   err_status (void);  /*ได้รับสถานะพอร์ต*/
void  write_ready (void); /*ได้รับสถานะรีจิสเตอร์การส่งข้อมูลและรีจิสเตอร์บัฟเฟอร์การส่งข้อมูล*/
int. read_ready (void);  /*การยืนยันสถานะการตั้งค่าข้อมูล*/
```

```
/*****************************************************************************/
/*                          การเขียนข้อมูล*/
/*****************************************************************************/
void  write_data (char wr_data);/*ข้อมูลถูกเขียนลงในรีจิสเตอร์*/
void  write (char *wr_data);*ข้อมูลถูกเขียนลงใน GP*/

/*****************************************************************************/
/*                          การอ่านข้อมูล*/
/*****************************************************************************/
int   read_data (void);/*ข้อมูลถูกอ่านจาก GP*/
void  change_screen (int interrupt_data);/*ระบุข้อมูลที่ได้รับในการอินเตอร์รัพต์จาก GP*/
int   read (void);/*อ่านข้อมูลที่ได้รับในการอินเตอร์รัพต์จาก GP*/

/*****************************************************************************/
/*                      การยืนยันข้อมูลที่ป้อนด้วยปุ่ม*/
/*****************************************************************************/
int   kbhit (void);

/*****************************************************************************/
/*                          ตัวแปรร่วม*/
/*****************************************************************************/
int   interrupt_data,port_status+;
char  *str2;

void main (void)
{
        int no_data;
        str2 =  (char*) malloc (sizeof (char) *data_sezi_str2); /*รักษาความปลอดภัยให้หน่วยความจำของ
str2*/
        char *wr_data =  (char*) malloc (sizeof (char) *data_size_wr_data);
                                        /*รักษาความปลอดภัยให้หน่วยความจำสำหรับ wr_data*/
        open_SIO ();          /*การตั้งค่าการสื่อสารสำหรับ RS232C*/
        wr_data = "\x1bW000F0001\x0d\0" ; /*เขียนข้อมูล 0x1 ลงในตำแหน่ง 15
ซึ่งจะเป็นการตั้งค่าหน้าจอหมายเลข 1*/
        write (wr_data);
        wr_data = "\x1bW0014003F\x0d\0";
              /เขียนข้อมูล *0x3F ในตำแหน่ง 20 ซึ่งจะเป็นการป้อนวัสดุลงในถังเติมอากาศหมายเลข 1 ถึง 6 */
        write (wr_data);
/*************************************************************************/
/*      ระบุการรับข้อมูลจาก GP  */
/*  ถ้ากดปุ่ม Write การดำเนินการจะเสร็จสมบูรณ์*/
/*************************************************************************/
       while (1)
       {
              no_data = read ();
```

```
            if (no_data == 1)      /*หากมีข้อมูลที่ป้อนด้วยปุ่มไม่ว่ากี่ข้อมูลก็ตาม, no_data=1*/
            {
                  break;
            }
             else
            {
                  wr_data = str2;
                  write (wr_data);
            }
      }
      getch ();                  /*รหัสสำหรับปุ่มจะถูกนำออกจากบัฟเฟอร์ปุ่ม*/
      free (wr_data);             /*พื้นที่หน่วยความจำสำหรับ wr_data จะถูกลบข้อมูลออก*/
      free (str2);                /*พื้นที่หน่วยความจำสำหรับ str2 จะถูกลบข้อมูลออก*/
}

/*ได้รับสถานะรีจิสเตอร์บัฟเฟอร์การส่งข้อมูลและสถานะรีจิสเตอร์การส่งข้อมูล*/
void write_ready (void)
{
      int err6000;

      err6000 = 0;
      while (status_bit_6000 != err6000)
      {
            err6000 = err_status () & status_bit_6000;
      }
      return;
}

/*การยืนยันสถานะการตั้งค่าข้อมูล*/
int read_ready (void)
{
      int no_data,err0020;

      err0020 = 0;
      while (status_bit_0020 != err0020)
      }
            err0020 = 344_status () & status bit_0020;
            if (kbhit ())       /*ยืนยันว่ามีข้อมูลที่ป้อนด้วยปุ่มหรือไม่*/
            {
                  no_data = 1;/*หากมีข้อมูลที่ป้อนด้วยปุ่มหนึ่งข้อมูล, no_data=1*/
                  break;      /*โปรแกรมสิ้นสุดการทำงาน*/
            }
```

```
    }
     return (no_data);
}
/*ข้อมูลถูกเขียนลงใน GP*/

void write (char *wr_data)
{
     while (*wr_data != '\0')        /*ข้อมูลถูกเขียนจนกว่าจะกลายเป็นศูนย์*/
    {
          write_ready ();
          write_data (*wr_data);
          wr_data++;          /*ตำแหน่งที่ชี้โดยตัวชี้จะเพิ่มขึ้น*/
     }
     return;
}
/*************************************************************************/
/*          ข้อมูลอินเตอร์รัพต์ที่ได้รับจาก GP ได้รับการยืนยัน        */
/*          ข้อมูลถูกเขียนลงในตำแหน่ง 20, 21, 22 และ 23         */
/*************************************************************************/
void change_screen (int interrupt_data)
{
     switch (interrupt_data)
/*หาก interrupt_data เป็น 1 จะเขียน 0x1 ในตำแหน่ง 21, 0x3F ในตำแหน่ง 22 และ 0x50 ในตำแหน่ง 23*/
          case   1: str2 = "\x1bW00150001003F0050\x0d\0";
                    break;

/*หาก interrupt_data เป็น 2 จะเขียน 0x2 ในตำแหน่ง 21, 0x0 ในตำแหน่ง 22 และ 0x0 ในตำแหน่ง 23*/
          case   2: str2 = "\x1bW0015000200000000\x9d\0";
                    break;

/*หาก interrupt_data เป็น 3 จะเขียน 0x4 ในตำแหน่ง 21, 0x0 ในตำแหน่ง 22 และ 0x50 ในตำแหน่ง 23*/
          case   3: str2 = "\x1bW00150000400000050\x0d\0";
                    break;

/*หาก interrupt_data เป็น 4 จะเขียน 0x8 ในตำแหน่ง 21, 0x0 ในตำแหน่ง 22 และ 0x20 ในตำแหน่ง 23*/
          case   4; str2 = "\x1bW0015000800000020\x0d\0";
                    break;

/*หาก interrupt_data เป็นค่าอื่น ๆ ที่ไม่ใช่ 1 ถึง 4 จะเขียน NULL*/
          default  : str2 = "\0";
                    break;
     {
     return;
```

```
    }

    /*************************************************************************/
    /*          อ่านข้อมูลอินเตอร์รัพต์จาก GP                */
    /*  ทำการอ่านข้อมูลจนกว่า interrupt_data จะกลายเป็นค่าอื่น ๆ ที่ไม่ใช่ NULL  */
    /*************************************************************************/
    int read (void)
    {
            int no_data;
            do
            {
                    no_data = read_ready (); /*การยืนยันสถานะการตั้งค่าข้อมูล*/
                    if (no_data == 1)       /*หากมีข้อมูลที่ป้อนด้วยปุ่มหนึ่งข้อมูล no_data=1*/
                    {
                         break;
                    }
                    else
                    {

                         read_data ();      /*อ่านข้อมูลที่ได้รับจาก GP/*
                         change_screen (interrupt_data); /*ระบุข้อมูลที่ได้รับจาก GP*/
                    }
             } while (*str2 == '\0');
             return (no_data);
    }
    /*การตั้งค่าการสื่อสารสำหรับ RS232C*/

void open_SIO (void)
{
        union REGS regs ;
                regs.x.dx = serial_port_number;
                regs.h.ah = serial_port_INT;
                regs.h.al = serial_port_parameter;
                int86 (serial_port_BIOS,&regs,&regs);
        return;
}

/*ได้รับสถานะพอร์ต*/
int err_status  (void)
{
        union REGS regs;
                regs.x.dx = serial_port_number;
                regs.h.ah = get_status;
```

```
            int86 (serial_port_BIOS,&regs,&regs);
            port_status = regs.x.ax;
        return (port_status);
}

/*ข้อมูลถูกเขียนลงในรีจิสเตอร์*/
void write_data (char wr_data)
{
        union REGS regs;
            regs.x.dx = serial_port number;
            regs.h.ah = serial_port_write;
            regs.h.al = wr_data;
            int86 (serial_port_BIOS,&regs,&regs);
        return;
}

/*ข้อมูลถูกอ่านจาก GP*/
int read_data (void)
{
        union REGS regs;
            regs.x.dx = serial_port_number;
            regs.h.ah = serial_port_read;
            int86 (serial_port_BIOS,&regs,&regs);
            interrupt_data = regs.h.al;
        return (interrupt_data);
}
```

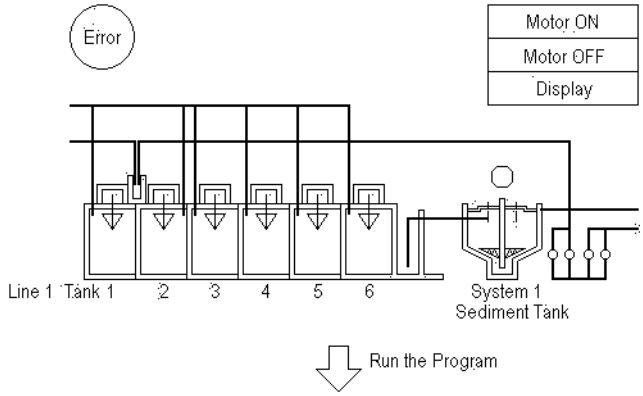
หมายเหตุ

- การใส่หรือไม่ใส่ open_SI0 (void), err_status (void), write_data (char wr_data) และ read_data (void) จะขึ้นอยู่กับรุ่นที่ใช้ หากเขียนโปรแกรมบนคอมพิวเตอร์ส่วนบุคคลที่ไม่ใช่ประเภท IBM-compatible จะต้องทำการแก้ไขเพื่อให้สามารถใช้งานได้

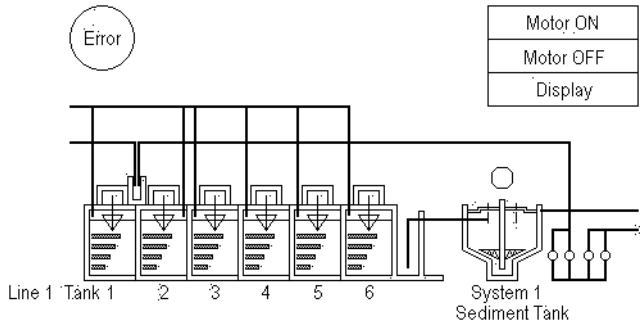
(4) หลังจากถ่ายโอนข้อมูลหน้าจอไปยัง GP แล้ว การแสดงผล (การทำงาน) จะเริ่มต้นขึ้น

◆ **หน้าจอเรียกใช้งานของ GP**

หน้าจอ GP (ก่อนเรียกใช้โปรแกรม)



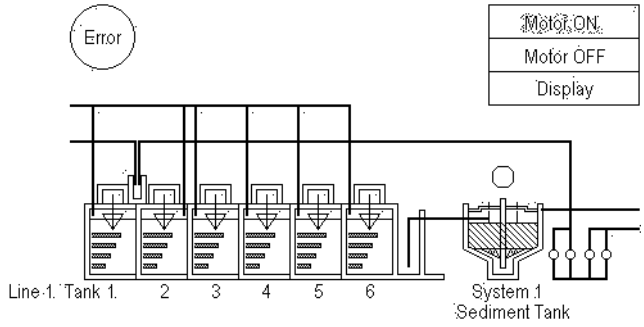
หน้าจอ GP (หลังเรียกใช้โปรแกรม)



ไลบรารีจำนวน 6 ไลบรารี จะปรากฏขึ้น

กดสวิตช์ [Motor ON]

รหัส ASCII "31" = ข้อมูล "1" ถูกส่งออกไปยังเครื่องโฮสต์ ทำให้หน้าจอเปลี่ยน



12.2 การแก้ปัญหาการสื่อสารของ GP (Multi-drop) หลายยูนิต

เครื่องโฮสต์จะทำหน้าที่สองอย่างด้วยกันเมื่อทำการควบคุมยูนิต GP หลายยูนิต ได้แก่:

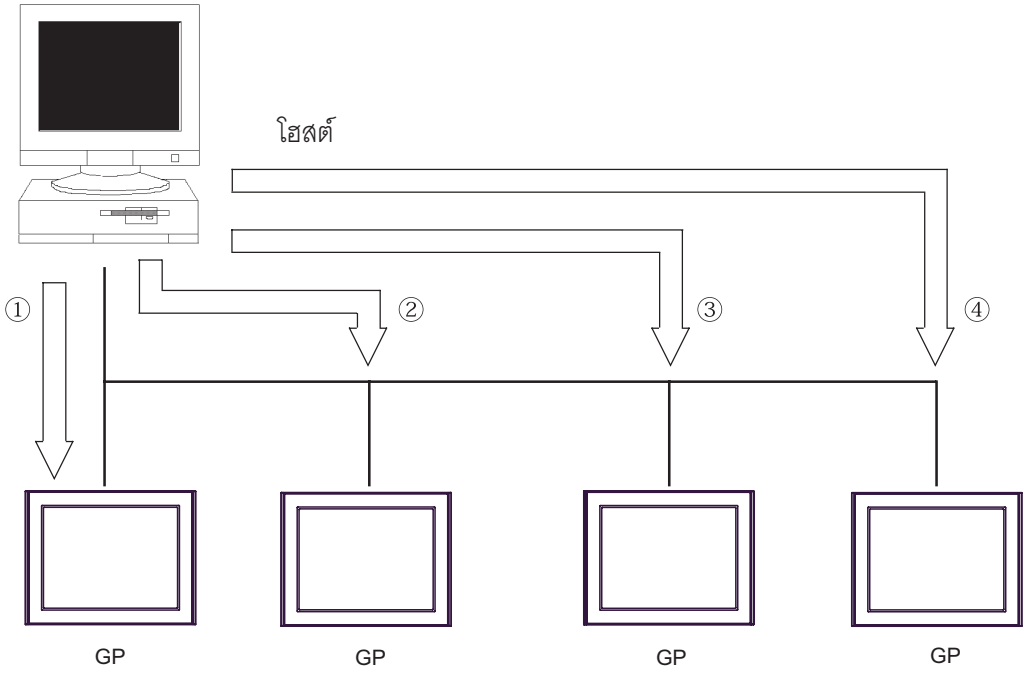
1. ถ่ายโอนข้อมูลที่จะแสดงผล
2. อ่านข้อมูลที่ป้อนโดยหน้าจอสัมผัสจากยูนิต GP ผ่านทางการโพล

โปรดทราบว่ายิ่งมีการควบคุมยูนิต GP เป็นจำนวนมากเท่าไร จำนวนข้อมูลที่จะถูกถ่ายโอนมากจะมีมากขึ้นเท่านั้น จึงทำให้เครื่องโฮสต์ต้องทำงานหนักขึ้นตามไปด้วย นอกจากนี้ จำนวนยูนิต GP หรือจำนวนข้อมูลที่มากเกินไปยังลดความเร็วในการตอบสนองของยูนิต GP ลง (เปลี่ยนการแสดงผลและตอบสนองต่อข้อมูลที่ป้อนโดยหน้าจอสัมผัสได้ช้าลง) และมีผลกระทบต่อการทำงานของระบบอย่างมาก ดังนั้น คุณควรพิจารณาจำนวนยูนิต GP และจำนวนข้อมูล
เมื่อทำการออกแบบระบบแบบ Multi-drop ด้วย

■ การส่งข้อมูลการแสดงผลไปยัง GP ทุกยูนิตพร้อมกัน

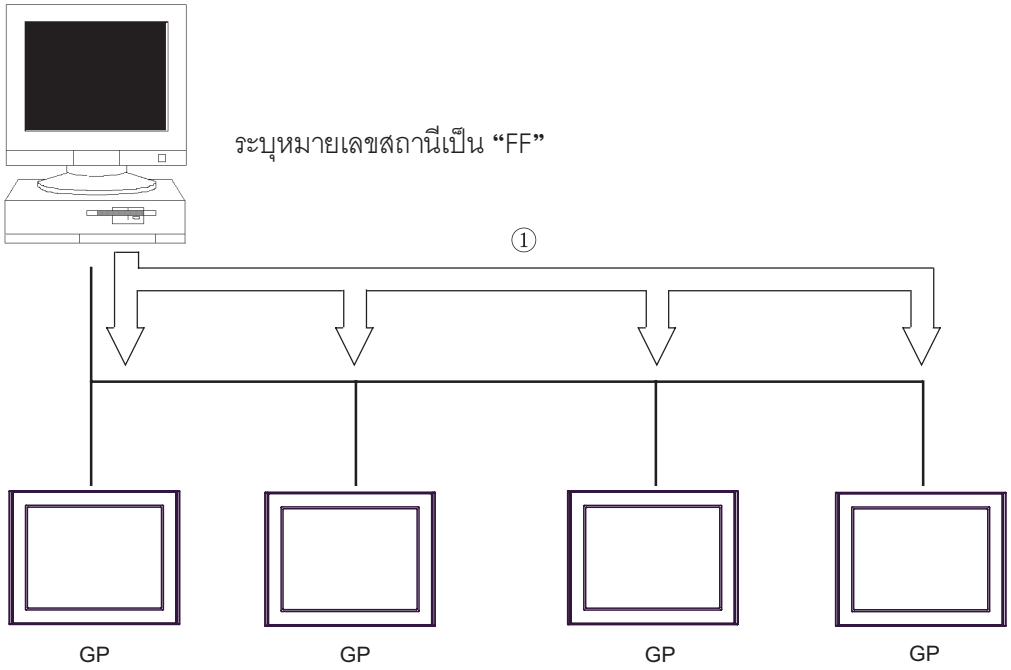
เมื่อคุณจำเป็นต้องส่งข้อมูลเดียวกันไปยัง GP ทุกยูนิต ให้ลองส่งข้อมูลไปยัง GP ทุกยูนิตพร้อมกัน
เพื่อเพิ่มประสิทธิภาพในการทำงาน
แทนที่จะส่งข้อมูลไปยัง GP ครั้งละหนึ่งยูนิต
(ทำได้โดยป้อนจำนวนสถานีเป็น "FF")

◆ **เมื่อส่งข้อมูลไปยัง GP ครั้งละหนึ่งยูนิต**



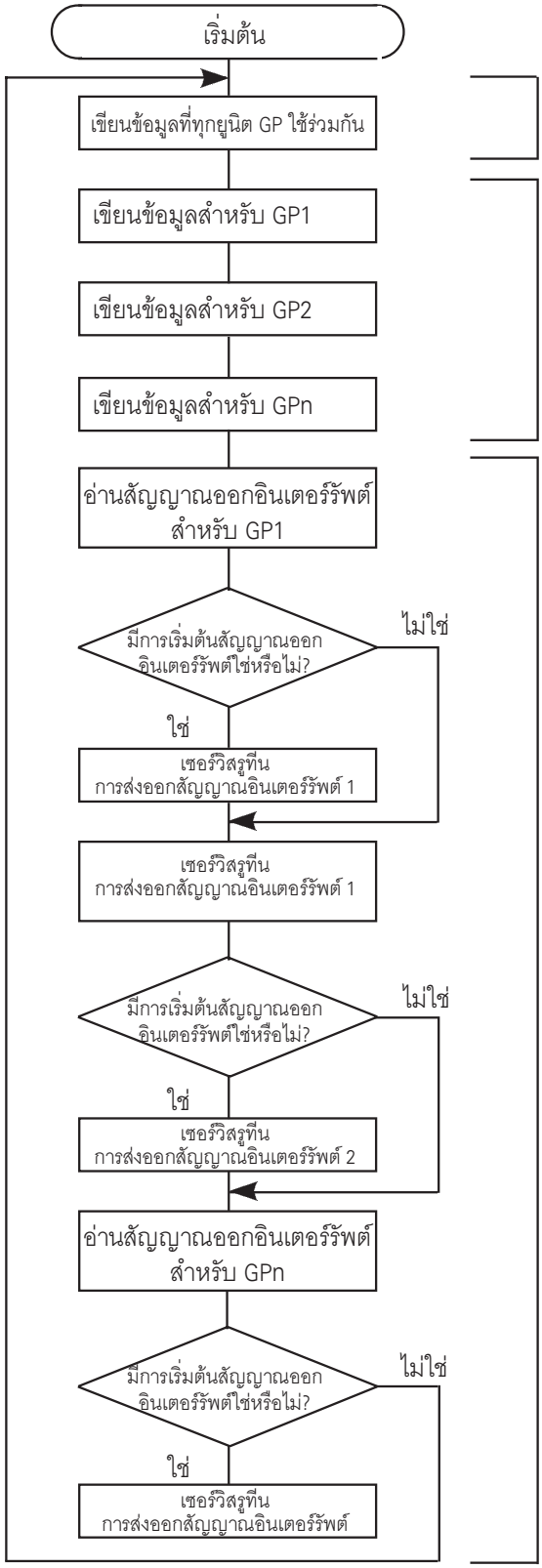
ใช้เวลานานมาก (นานกว่ากรณีที่แสดงต่อไปนี้ 4 เท่า)

◆ **เมื่อส่งข้อมูลไปยัง GP ทุกยูนิตพร้อมกัน**



ใช้เวลาลดลงมาก (น้อยกว่ากรณีที่แสดงข้างต้น 4 เท่า)

12.3 แผนผังการทำงานของโปรแกรมสำหรับระบบ Multi-drop



(1) เขียนข้อมูลที่ GP ทุกยูนิตใช้ร่วมกัน
ใช้คำสั่ง ESC W เพื่อเขียนข้อมูลการแสดงผลที่ GP ทุกยูนิตใช้ร่วมกันลงในพื้นที่ระบบ
(ในขั้นนี้ ให้ระบุจำนวนสถานีเป็น "FF")
(2) เขียนข้อมูลสำหรับยูนิต GP ที่ระบุ
ใช้คำสั่ง ESC W เพื่อเขียนข้อมูลสำหรับยูนิต GP ที่ระบุลงในพื้นที่ระบบ

(3) ทำการโพล
ใช้คำสั่ง ESC I เพื่อทำการโพลแต่ละยูนิต GP เพื่อพิจารณาว่ามีการป้อนข้อมูลทางหน้าจอสัมผัสหรือไม่ จากนั้นจะใช้ข้อมูลที่ป้อนโดยหน้าจอสัมผัสตามลำดับ

หมายเหตุ

- ในการเพิ่มความเร็วในการตอบสนองของ GP ต่อข้อมูลที่ป้อนโดยหน้าจอสัมผัส ให้แทรกลำดับการโพลหลังจากลำดับการเขียนแต่ละลำดับ (ลำดับที่ข้อมูลถูกเขียนของยูนิต GP ที่ระบุ)
- จำนวนข้อมูลที่จะเขียนในพื้นที่ระบบต้องมีจำนวนน้อยที่สุดเท่าที่เป็นไปได้
  เช่น คุณอาจเลือกอัพเดตเฉพาะรายการข้อมูลที่มีการปลี่ยนแปลงเท่านั้น

13 โปรแกรมตัวอย่าง (การสื่อสารแบบ Ethernet)

ในบทนี้จะอธิบายเกี่ยวกับโปรแกรมตัวอย่าง (AGPM.EXE) ที่ใช้ LAN API แบบเชื่อมต่อผ่านหน่วยความจำที่มีอยู่ในยูนิต GP Ethernet I/F โปรแกรม AGPM.EXE เป็นโปรแกรมตัวอย่างที่ทำให้สามารถเข้าใช้หน่วยความจำใน GP ในแบบเรียลไทม์จากเครื่องโฮสต์ ผ่านการเชื่อมต่อแบบ 1:1 หรือ 1:n (มัลติลิงค์) ระหว่างยูนิต GP กับเครื่องโฮสต์ที่ใช้ระบบปฏิบัติการ Windows

13.1 โปรแกรมตัวอย่างของ LAN API แบบเชื่อมต่อผ่านหน่วยความจำ

■ สภาพแวดล้อมในการเริ่มทำงาน

(1) โปรแกรม AGPM.EXE ทำงานบนระบบปฏิบัติการ Windows 98

(2) เนื่องจากโปรแกรม AGPM.EXE ใช้ไฟล์ MtoMLAN.DLL ให้คัดลอกไฟล์ MtoMLAN.DLL ลงในโฟลเดอร์ Windows

(3) LAN แบบเชื่อมต่อผ่านหน่วยความจำจะใช้โปรโตคอล TCP/IP คุณจึงต้องติดตั้งโปรโตคอล TCP/IP ก่อน (ติดตั้ง Microsoft TCP/IP โดยเลือก [Start] - [Control Panel] - [Network])

หมายเหตุ
- หากข้อความ "DLL: LAN initialize error" ปรากฏขึ้นขณะเริ่มต้นโปรแกรม AGPM.EXE และโปรแกรมไม่สามารถเริ่มทำงานได้ อาจมีสาเหตุมาจากการตั้งค่า TCP/IP ไม่ถูกต้อง ให้ตรวจสอบการตั้งค่า TCP/IP

■ สภาพแวดล้อมในการพัฒนา

โปรแกรม AGPM.EXE ได้ถูกพัฒนาขึ้นเพื่อใช้งานในสภาพแวดล้อมต่อไปนี้:
ซอร์สโค้ดของโปรแกรมตัวอย่างมีอยู่ในซีดีรอม GP-Pro EX
เมื่อมีการแปลซอร์สโค้ดที่อยู่ในโฟลเดอร์ [MTOMLAN] ของซีดีรอมในสภาพแวดล้อมต่อไปนี้ ไฟล์ [AGPM.EXE] จะถูกสร้างขึ้น

คอมไพเลอร์ : Microsoft Visual C++ Ver 6.0
ระบบปฏิบัติการ : Microsoft Windows 98

■ วิธีเข้าใช้ API แบบเชื่อมต่อผ่านหน่วยความจำ

โปรแกรม AGPM.EXE จะกำหนดและใช้ Class CMSock เพื่อเข้าใช้ API แบบเชื่อมต่อผ่านหน่วยความจำ
Class CMSock จะรวม API แบบเชื่อมต่อผ่านหน่วยความจำเป็น "หนึ่งเต้ารับ - หนึ่งออบเจ็กต์"
โปรแกรม AGPM.EXE จะทำการติดต่อกลับจาก API แบบเชื่อมต่อผ่านหน่วยความจำด้วยการลบวิธี Class CMSock

■ ที่มาของ CMSock

โปรแกรม AGPM.EXE จะใช้คลาสสองคลาสที่ได้รับมาจาก CMSock โดยคลาสแรกจะได้มาจาก Class CGpMApp สำหรับการเชื่อมต่อแบบ 1:n และการค้นหาโหนด และคลาสที่สองจะได้มาจาก Class CGpMDoc สำหรับการเชื่อมต่อแบบ 1:1

CGpMApp เป็นคลาสแอ็พพลิเคชันของ AGPM และ CGpMDoc เป็นคลาสเอกสารของ AGPM กล่าวอีกนัยหนึ่งคือคลาสแอ็พพลิเคชันจะจัดการการเชื่อมต่อแบบ 1:n และคลาสเอกสารจะจัดการการเชื่อมต่อแบบ 1:1

■ Class CGpMDOC

คลาสนี้คือแกนหลักของโปรแกรม AGPM.Exe โดยใช้สำหรับการจัดการข้อมูลเอกสาร รวมถึงกลุ่มออบเจ็กต์ข้อมูล นอกจากนี้ คลาสนี้ยังจัดการการเชื่อมต่อแบบ 1:1 กับยูนิต GP อีกด้วย

■ Class CGpMView

คลาสนี้จะแสดงออบเจ็กต์ข้อมูลที่เกี่ยวข้องของคลาส CGpMDoc ในหน้าต่าง

■ MtoMAPI.H และ MtoMLAN.LI

โปรแกรม AGPM.EXE มีไฟล์ MtoMAPI.H ของอุปกรณ์ภายนอก
โดยไฟล์ MtoMAPI.H จะถูกบันทึกไว้ในโฟลเดอร์ [MtoMLAN] ให้คัดลอกไฟล์นี้ลงในโฟลเดอร์ที่เหมาะสม และระบุตำแหน่งโดยเปลี่ยนข้อความคำสั่ง #include ของ defsfile.h

โปรแกรม AGPM.EXE มีไฟล์ MtoMLAN.LIB สำหรับเรียกโปรแกรม MtoMLAN.DLL ให้คัดลอกไฟล์นี้ลงในโฟลเดอร์ที่เหมาะสม และระบุตำแหน่งโดยเลือก [Setup] - [Linker] - [Object/Library Module]