

20

การใช้สคริปต์

(การเขียนโปรแกรมโดยไม่ใช้พาร์ท)

บทนี้จะอธิบายเกี่ยวกับหลักการพื้นฐานของการสร้างและการใช้สคริปต์เขียนโปรแกรมใน GP-Pro EX โปรดเริ่มต้นด้วยการอ่าน “20.1 เมนูการตั้งค่า” (หน้า 20-2) แล้วจึงไปอ่านหน้าที่เกี่ยวข้อง

20.1	เมนูการตั้งค่า.....	20-2
20.2	การทำงานตามเงื่อนไข.....	20-5
20.3	การคัดลอกข้อมูลในบล็อก.....	20-12
20.4	การแสดงการแจ้งเตือนเมื่อเกิดข้อผิดพลาด.....	20-17
20.5	การสื่อสารกับอุปกรณ์ต่อพ่วงที่ไม่รองรับ.....	20-21
20.6	ขั้นตอนการสร้างสคริปต์.....	20-41
20.7	การตั้งค่าเงื่อนไขการทริกเกอร์.....	20-46
20.8	คำแนะนำในการตั้งค่า.....	20-53
20.9	ข้อจำกัด.....	20-59


20.1 เมนูการตั้งค่า

คุณสามารถใช้ D-Scripts ในการสร้างโปรแกรมที่ไม่ซับซ้อนได้ คุณสามารถใช้คุณสมบัตินี้เพื่อทำงานใน GP หรือเพื่อสื่อสารระหว่าง GP กับอุปกรณ์ต่อพ่วงที่ไม่รองรับได้

คำเตือน

ห้ามใช้ D-Scripts/Global D-Scripts ควบคุมระบบการทำงานที่อาจก่อให้เกิดอันตรายถึงแก่ชีวิตหรือสร้างความเสียหายร้ายแรง

หมายเหตุ

- การตั้งค่า D-Scripts จะทำบนหน้าจอหลัก โดยหน้าจอหลักดังกล่าวจะพิจารณาเงื่อนไขขณะที่หน้าจอกำลังแสดงอยู่แล้วจึงเรียกใช้สคริปต์นั้น
 - เมื่อ GP กำลังทำงาน Global D-Script จะรันตามทริกเกอร์ ไม่ว่าหน้าจอที่แสดงจะเป็นอย่างไรก็ตาม
 - Extended Script ใช้สำหรับโปรแกรมการสื่อสารระดับสูง
 - นอกจากสคริปต์แล้ว คุณยังสามารถใช้ลอจิกโปรแกรมเพื่อการควบคุมได้
-  “28.1 เมนูการตั้งค่า” (หน้า 28-2)

การทำงานตามเงื่อนไข


สร้างสคริปต์ซึ่งจะเปลี่ยนหน้าจอต่างๆ เป็นหน้าจอหมายเลข 7 หลังจากเวลาผ่านไป 3 วินาที

หลังจาก 1 วินาที

เวลา →

ประมวลผลสคริปต์ → ประมวลผล → ประมวลผล → ประมวลผล


D100=1



D100 ไม่ใช่ 3 ดังนั้นคำสั่งที่อยู่หลัง "if" จะไม่ทำงาน

หลังจาก 2 วินาที


D100=2



D100 ไม่ใช่ 3 ดังนั้นคำสั่งที่อยู่หลัง "if" จะไม่ทำงาน

หลังจาก 3 วินาที

D100=3



D100 = 3 ดังนั้นเงื่อนไขเป็นจริงและ (w:LSD008)=7 จะทำงาน

☞ ขั้นตอนการตั้งค่า (หน้า 20-6)

☞ ข้อมูลเบื้องต้น (หน้า 20-5)


การคัดลอกข้อมูลในบล็อก

สร้างสคริปต์เพื่อตรวจสอบขาขึ้น (0 เปลี่ยนเป็น 1) ของตำแหน่งบิต M0100 แล้วคัดลอกข้อมูลที่จัดเก็บไว้ในอุปกรณ์เชื่อมต่อไปไว้ที่ตำแหน่งอื่น

D0099

C
·
·
·
B
A

D0000




➔

D0200

C
·
·
·
B
A

D0101



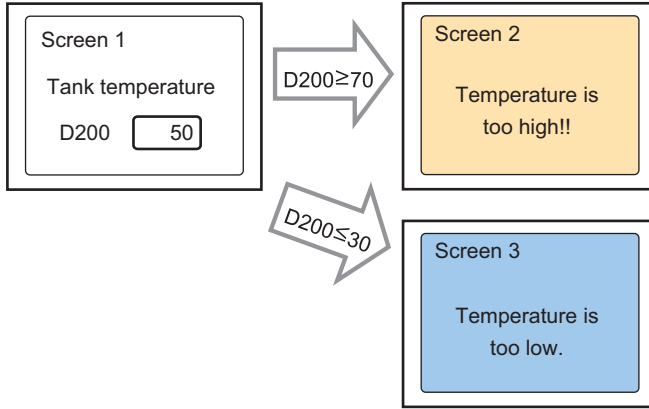
☞ ขั้นตอนการตั้งค่า (หน้า 20-13)

☞ ข้อมูลเบื้องต้น (หน้า 20-12)

การแสดงการแจ้งเตือนเมื่อเกิดข้อผิดพลาด

ระบบจัดการอุณหภูมิจะตรวจหาบิตที่มีข้อผิดพลาดจากอุปกรณ์ที่เชื่อมต่อไว้ และแสดงข้อความแจ้งเตือนเมื่อตำแหน่งจัดเก็บข้อมูลอุณหภูมิ (D200) มีค่าเพิ่มขึ้นถึงระดับตั้งแต่ 70°C ขึ้นไป หรือมีค่าลดลงถึงระดับตั้งแต่ 30°C ลงไป โดยสคริปต์นี้จะนับจำนวนข้อผิดพลาดที่ตรวจพบด้วย

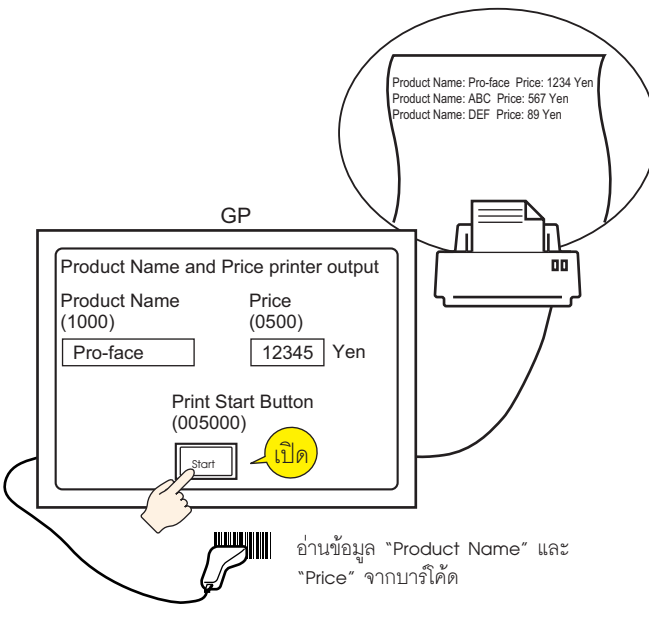
- ☞ ขั้นตอนการตั้งค่า (หน้า 20-18)
- ☞ ข้อมูลเบื้องต้น (หน้า 20-17)



การสื่อสารกับอุปกรณ์ต่อพ่วงที่ไม่รองรับ

สร้าง Extended Script เพื่อส่งข้อมูลที่อ่านได้จากบาร์โค้ดซึ่งเชื่อมต่อกับพอร์ต USB ไปยังเครื่องพิมพ์ที่ต่อกับพอร์ตCOM1 แบบอนุกรม

- ☞ ขั้นตอนการตั้งค่า (หน้า 20-37)
- ☞ ข้อมูลเบื้องต้น (หน้า 20-21)



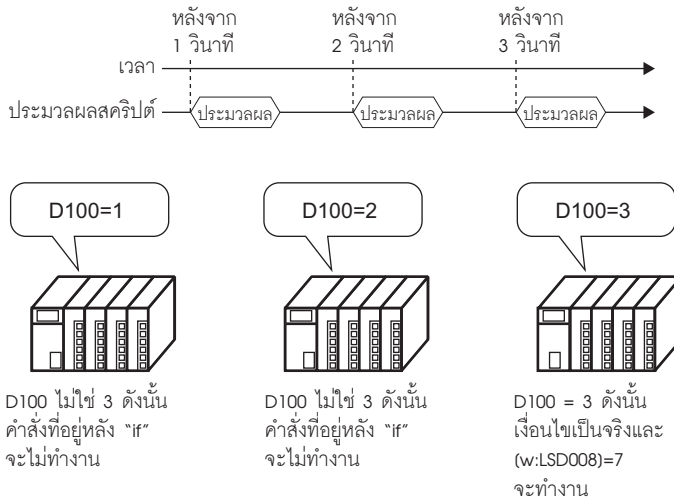
20.2 การทำงานตามเงื่อนไข

หมายเหตุ

- โปรดอ่านรายละเอียดจากคำแนะนำในการตั้งค่า
☞ “20.8.1 คำแนะนำในการตั้งค่า D-Script/[Global D-Script] ทั่วไป” (หน้า 20-53)

การดำเนินการ

สร้างสคริปต์ซึ่งจะเปลี่ยนหน้าจอต่าง ๆ เป็นหน้าจอหมายเลข 7 โดยอัตโนมัติ หลังจากเวลาผ่านไป 3 วินาที

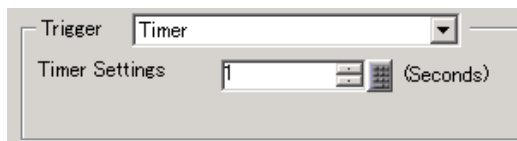


คำสั่งที่ใช้

คำสั่ง	ข้อมูลสรุปของฟังก์ชัน
Assignment (=)	กำหนดค่าฝั่งขวาให้แก่ค่าฝั่งซ้าย
Addition (+)	เพิ่มค่าคงที่ให้แก่ข้อมูลของอุปกรณ์ชนิดเวิร์ด
if ()	เมื่อเงื่อนไขเป็นจริง ระบบจะดำเนินการกระบวนการที่อยู่หลังข้อความคำสั่ง “if ()”
Equivalent (==)	เปรียบเทียบค่าฝั่งซ้ายและขวา เงื่อนไขจะเป็นจริงหากฝั่งซ้ายเท่ากับฝั่งขวา
LS0008	เปลี่ยนเป็นหมายเลขหน้าจอที่จัดเก็บไว้ในคำนี้ ☞ “A.1.4.2 พื้นที่เก็บข้อมูลระบบ” (หน้า A-11)

เงื่อนไขการทริกเกอร์

เลือก Timer ตามภาพด้านล่างนี้ แล้วตั้งค่า [Timer Settings] เป็น 1 วินาที



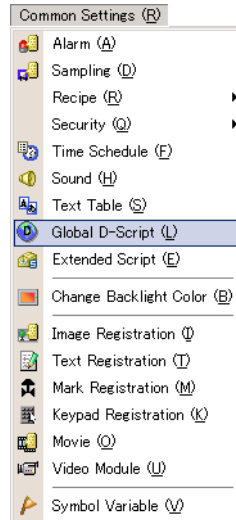
สคริปต์ที่เสร็จแล้ว

```

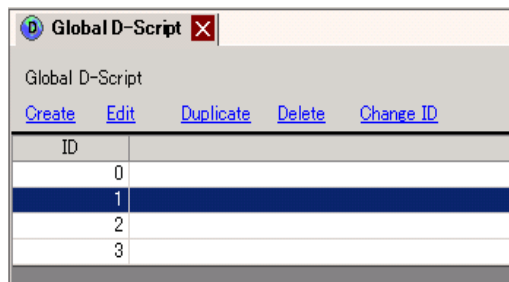
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 [w:[PLC1]D00100]=[w:[PLC1]D00100]+1
0002 if ([w:[PLC1]D00100]==3)
0003 {
0004   [w:[#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
    
```

ขั้นตอนการสร้าง

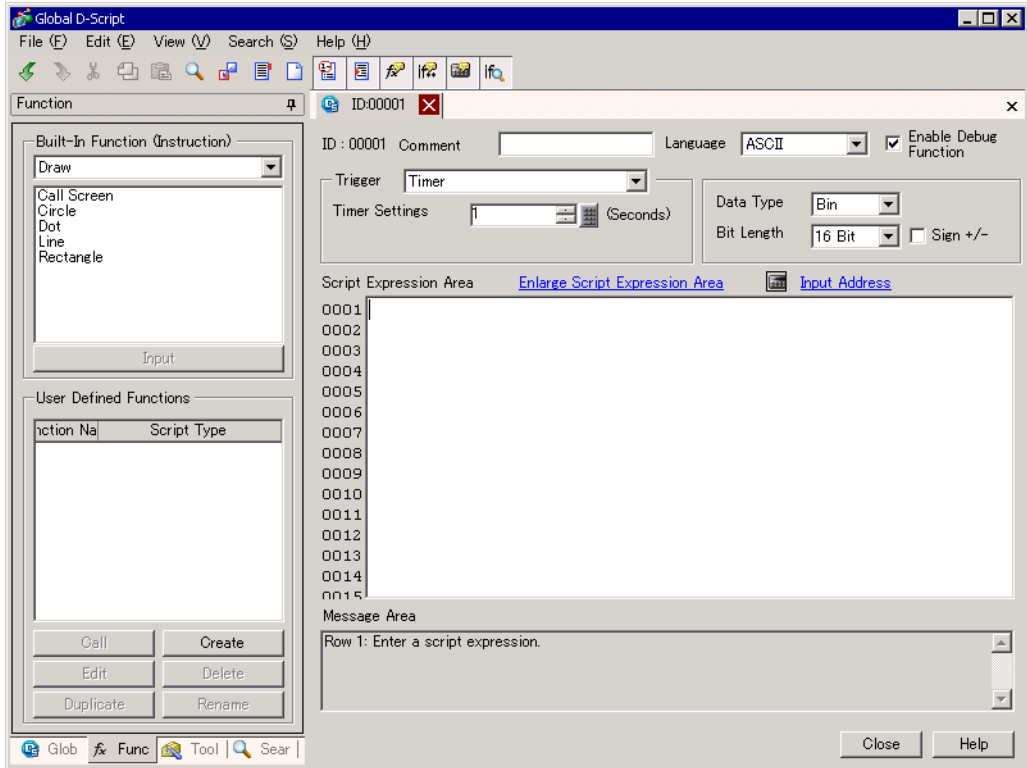
- 1 จากเมนู [Common Settings (R)] ให้เลือก [Global D-Script Settings (L)]



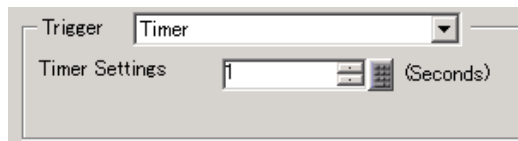
- 2 คลิก [Create] หากต้องการดูสคริปต์ที่มีอยู่ ให้เลือกหมายเลข ID และคลิกที่ [Edit] หรือดับเบิลคลิกที่แถวหมายเลข ID



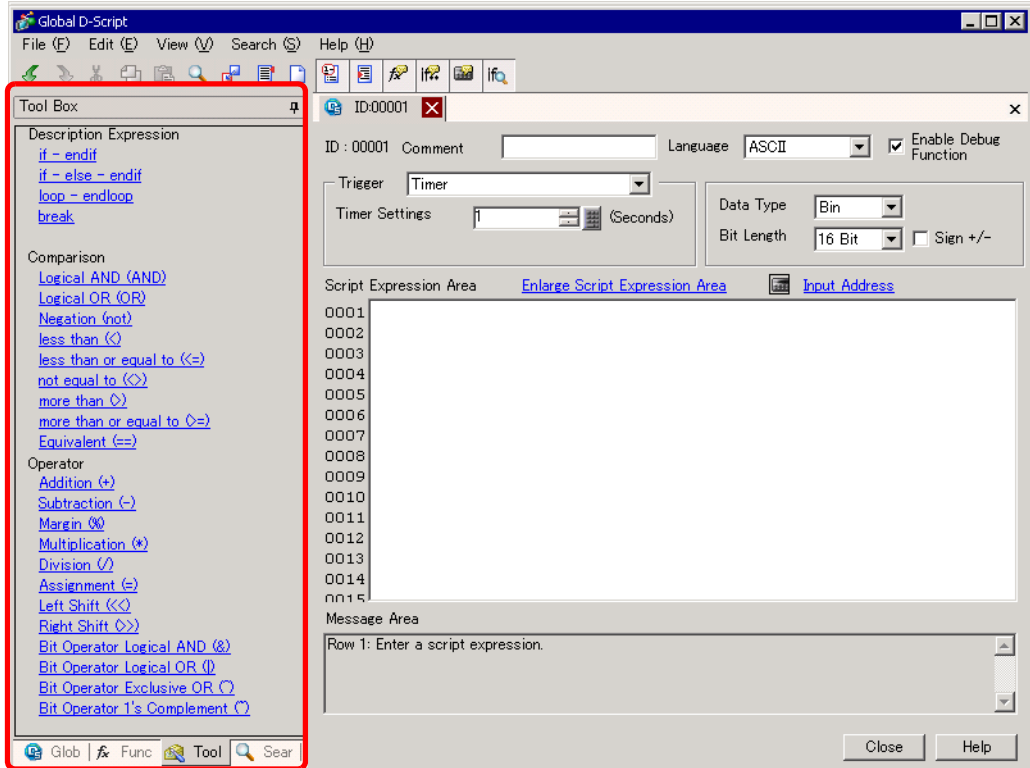
3 กล้องโต้ตอบ [Global D-Script] จะปรากฏขึ้น



4 เลือก [Timer] ในเงื่อนไขการทริกเกอร์สคริปต์ (trigger) และเลือก [Timer] แล้วระบุ [Timer Settings] เป็น 1 วินาที

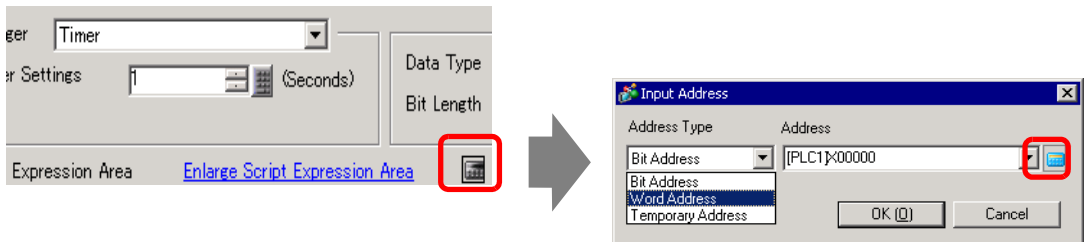


5 คลิกแท็บ [Tool Box] กล่องเครื่องมือนี้ช่วยให้คุณใส่คำสั่งที่จะใช้ในสคริปต์ได้อย่างง่ายดาย

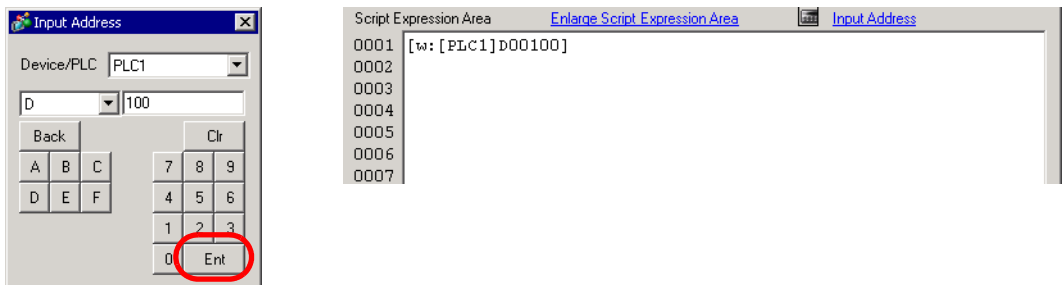


6 สร้างสคริปต์บรรทัดแรก หากคุณระบุค่าดีฟอลต์ D00100 เป็น 0 การดำเนินการในบรรทัดแรกจะเป็นการนับ ซึ่งจะเพิ่มขึ้นและจัดเก็บค่าที่นับได้ทุกครั้งที่กระบวนการเสร็จสิ้น

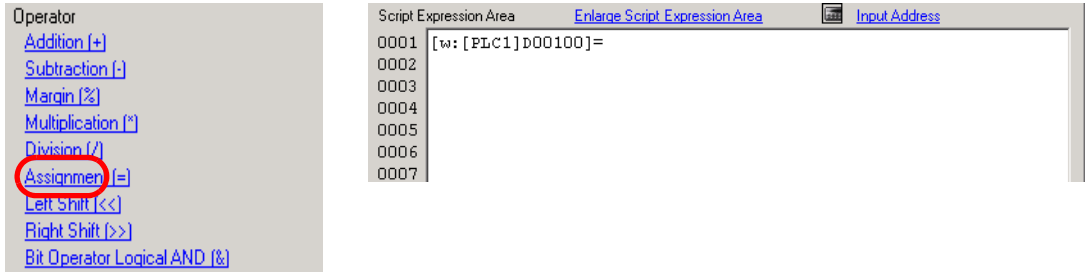
คลิกที่ และเลือก [Word Address] แล้วคลิกที่



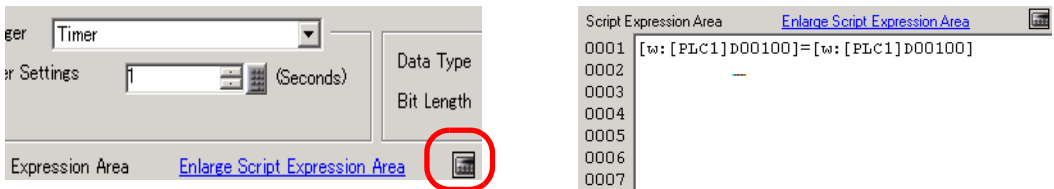
7 ป้อน D00100 แล้วคลิก [ENT]



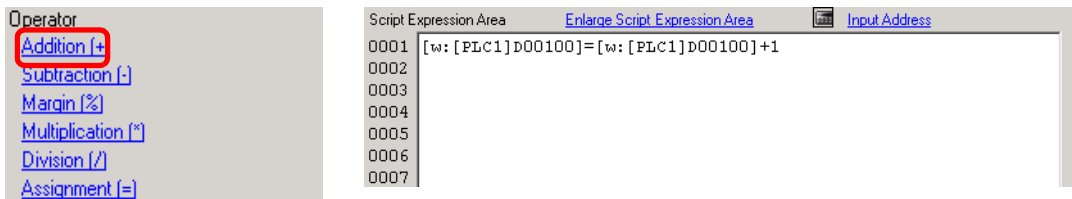
8 คลิก [Assignment (=)] ในกล่องเครื่องมือ



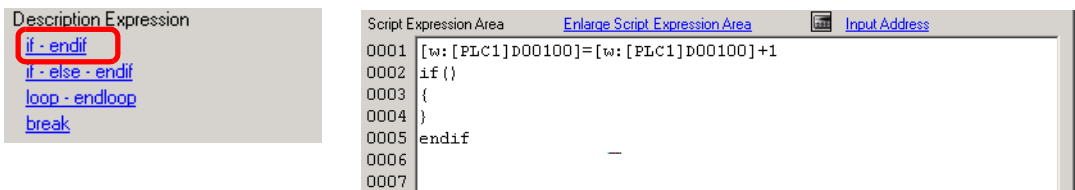
9 ใส่ D00100 โดยทำเหมือนกับขั้นตอนที่ 6-7



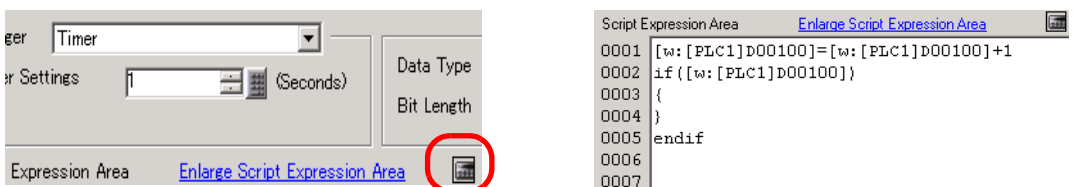
10 คลิกที่ [Addition (+)] และพิมพ์ “1” แถวแรกเสร็จสมบูรณ์แล้ว



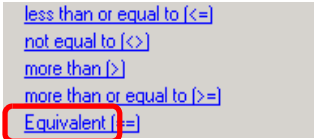
11 สร้างสคริปต์แถวที่สอง ในแถวที่สอง เมื่อเงื่อนไขเป็นจริง ระบบจะดำเนินการกระบวนการที่อยู่หลังข้อความคำสั่ง “if ()” คลิก [if - endif]



12 สร้างนิพจน์เงื่อนไขในวงเล็บ “()” ที่อยู่หลัง “if” นิพจน์เงื่อนไขนี้จะเปรียบเทียบค่าที่เก็บไว้ใน D00100 กับ “3” และเงื่อนไขจะเป็นจริงหากค่าเท่ากัน เลื่อนเคอร์เซอร์ให้อยู่ในวงเล็บ “()” แล้วทำซ้ำขั้นตอนที่ 6 ถึง 7 เพื่อใส่ D00100 อีกหนึ่งครั้ง

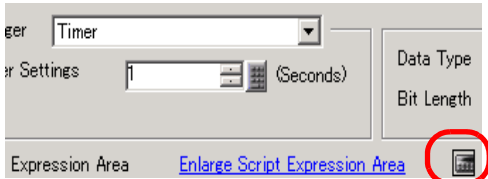


13 คลิก [Equivalent (==)] แล้วป้อน “3” แถวที่สองเสร็จสมบูรณ์แล้ว



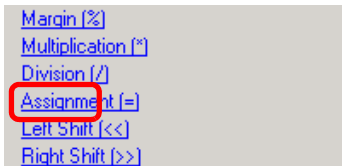
```
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 [w: [PLC1]D00100]=[w: [PLC1]D00100]+1
0002 if ([w: [PLC1]D00100]==3)
0003 {
0004 }
0005 endif
0006
0007
```

14 เลื่อนเคอร์เซอร์ให้อยู่ในวงเล็บ “{ }” แล้วกด Enter ทำซ้ำขั้นตอนที่ 6 ถึง 7 เพื่อใส่ LS0008 อีกครั้งหนึ่ง



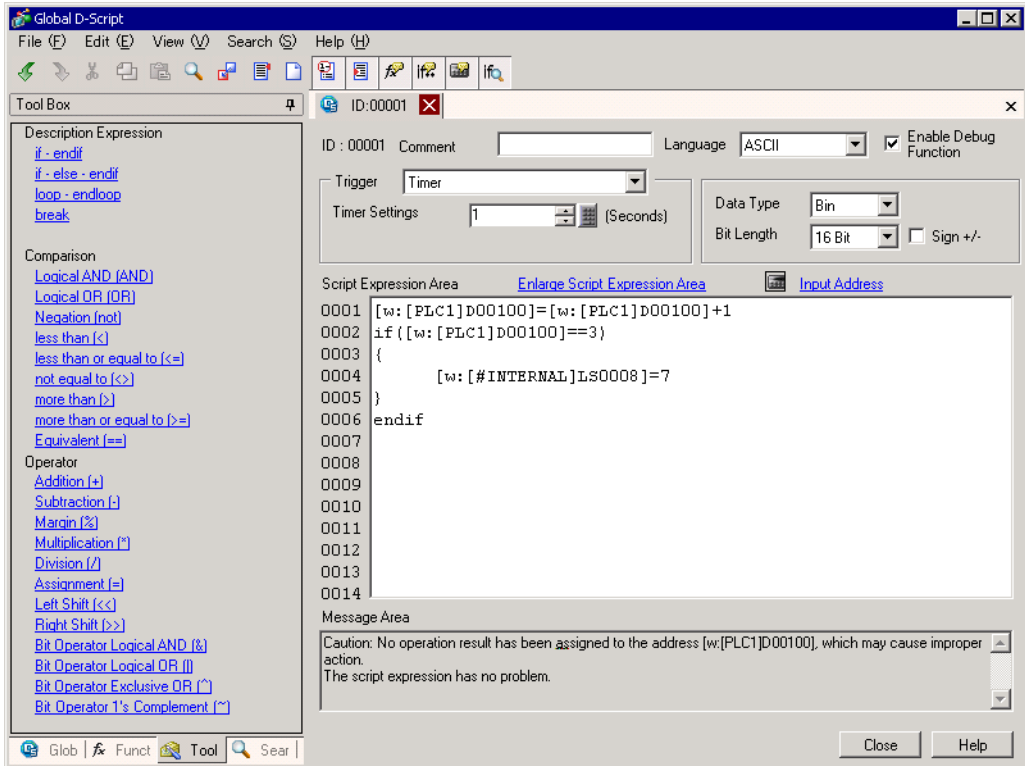
```
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 [w: [PLC1]D00100]=[w: [PLC1]D00100]+1
0002 if ([w: [PLC1]D00100]==3)
0003 {
0004     [w: [#INTERNAL]LS0008]
0005 }
0006 endif
0007
```

15 คลิก [Assignment (=)] แล้วป้อน “7”



```
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 [w: [PLC1]D00100]=[w: [PLC1]D00100]+1
0002 if ([w: [PLC1]D00100]==3)
0003 {
0004     [w: [#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
```

16 สคริปต์นี้เสร็จสมบูรณ์แล้ว



หมายเหตุ

- เมื่อเลือกข้อความ ให้กดปุ่ม [Ctrl] + [Shift] + [->] หรือ [-<] เพื่อเลือกข้อความทั้งบรรทัด
- กดปุ่ม [Ctrl] + [F4] เพื่อปิดการแสดงผลที่เลือกในปัจจุบัน
- กดปุ่ม [Esc] เพื่อเขียนทับสคริปต์ หรือลบและออกจากการทำงาน

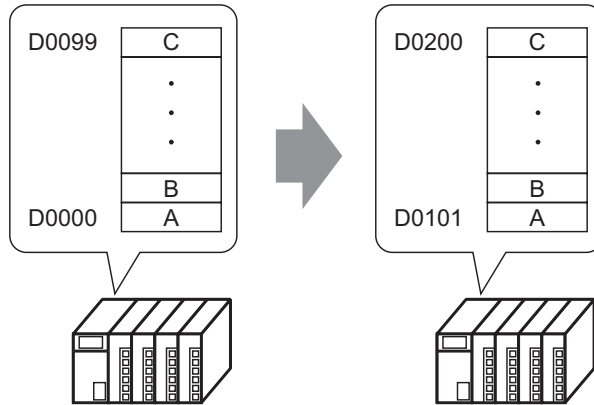
20.3 การคัดลอกข้อมูลในบล็อค

หมายเหตุ

- โปรดอ่านรายละเอียดจากคำแนะนำในการตั้งค่า
☞ “20.8.1 คำแนะนำในการตั้งค่า D-Script/[Global D-Script] ทั่วไป” (หน้า 20-53)

การดำเนินการ

สร้างสคริปต์เพื่อตรวจสอบหาขอบขาขึ้น (0 เปลี่ยนเป็น 1) ของตำแหน่งบิต M0100 แล้วคัดลอกข้อมูล
ที่จัดเก็บไว้ในอุปกรณ์เชื่อมต่อไปไว้ที่ตำแหน่งอื่น

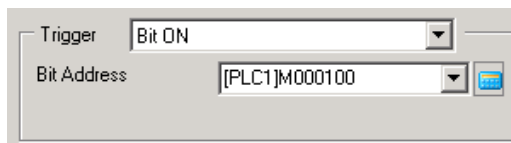


คำสั่งที่ใช้

คำสั่ง	ข้อมูลสรุปของฟังก์ชัน
Copy Memory memcpy ()	<p>คัดลอกค่าที่เก็บไว้ในอุปกรณ์โดยทำเพียงครั้งเดียว</p> <p>ระบบจะคัดลอกข้อมูลจำนวนตำแหน่งไปยังตำแหน่งเวิร์ดของปลายทางการคัดลอก โดยเริ่มจากตำแหน่งเวิร์ดแรกสุดของข้อมูลต้นทาง</p> <p>[Format]</p> <p>memcpy ([Copy To Address], [Copy From Address], Words)</p>

เงื่อนไขการทริกเกอร์

ใน [Trigger Condition] ให้เลือก [Bit ON] และกำหนด [Bit Address] เป็น M000100




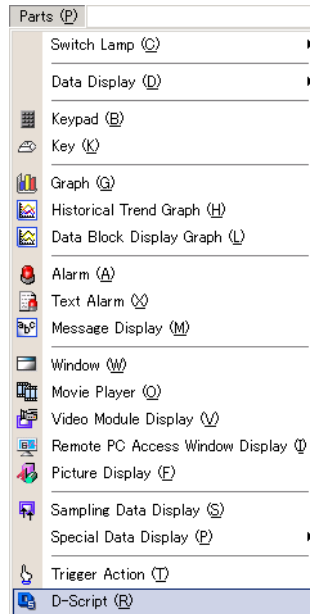
สคริปต์ที่เสร็จแล้ว

```

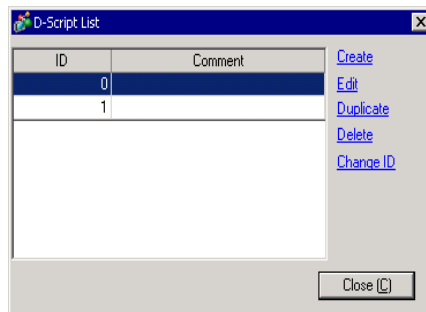
Script Expression Area
Enlarge Script Expression Area
Input Address
0001 memcpy ([w:[PLC1]D00101], [w:[PLC1]D00000], 100)
0002
0003
0004
0005
0006
0007
    
```

ขั้นตอนการสร้าง

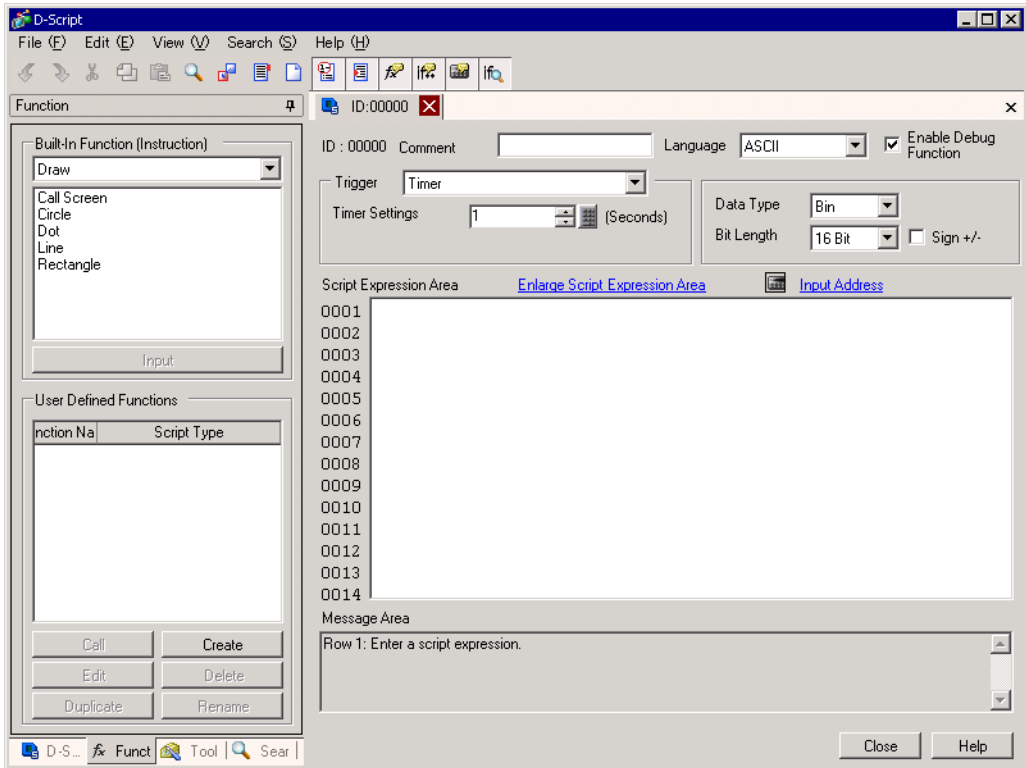
1 จากเมนู [Parts (P)] ให้เลือก [D-Script (R)] หรือคลิกที่  ในแถบเครื่องมือ



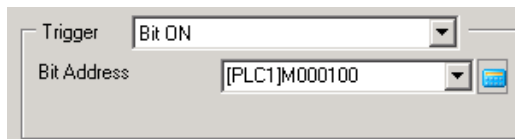
2 คลิก [Create] ID ของสคริปต์ที่มีอยู่จะแสดงอยู่ใน [D-Script List]



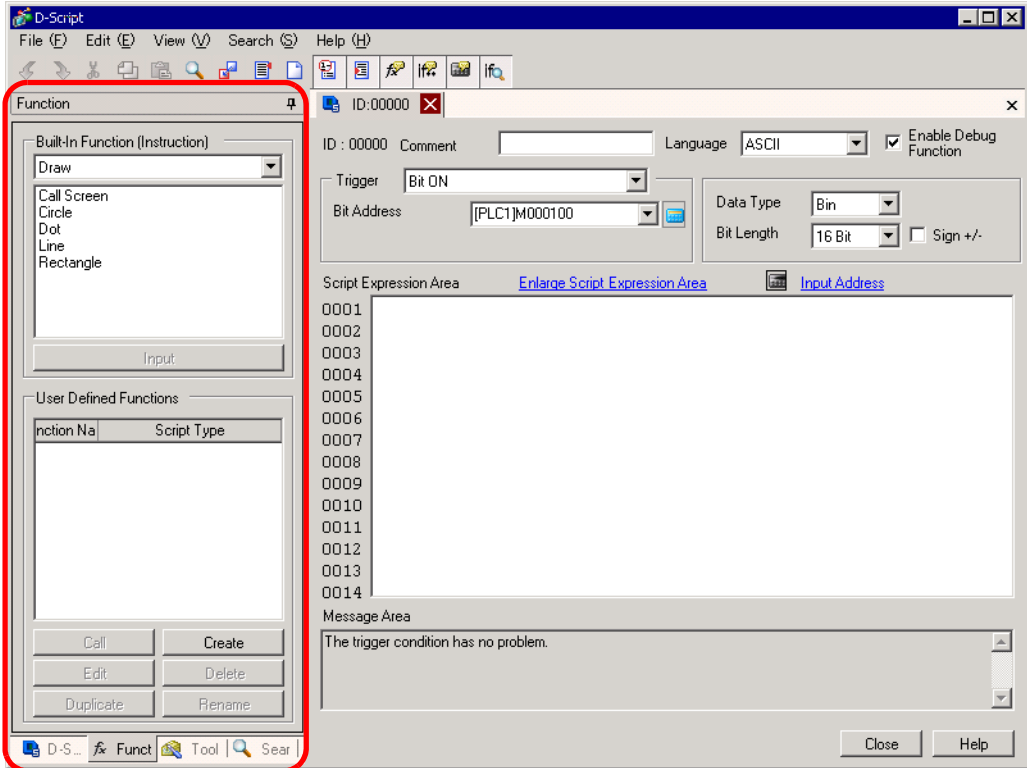
3 กล้องโต้ตอบ [D-Script] จะปรากฏขึ้น



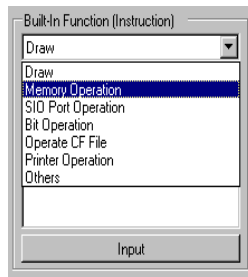
4 เลือก [Bit ON] ในเงื่อนไขการทริกเกอร์สคริปต์ (trigger) และระบุ M000100 เป็น [Bit Address]




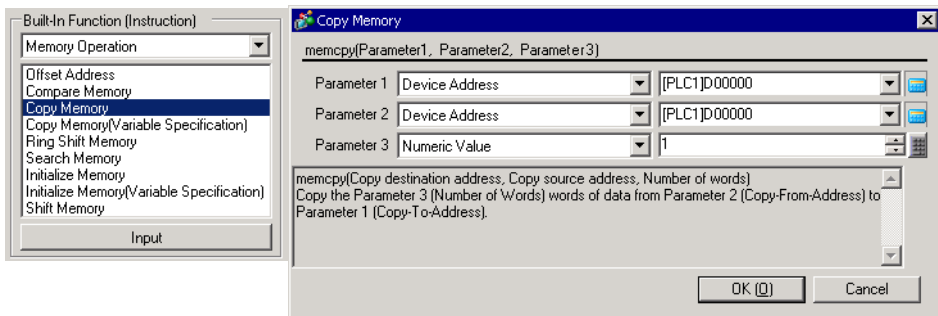
5 คลิกที่ [Function] คุณสามารถใส่คำสั่งที่ใช้ในสคริปต์ได้อย่างง่ายดายเพียงแค่คลิกที่ฟังก์ชันต่างๆ ที่มีให้เลือก



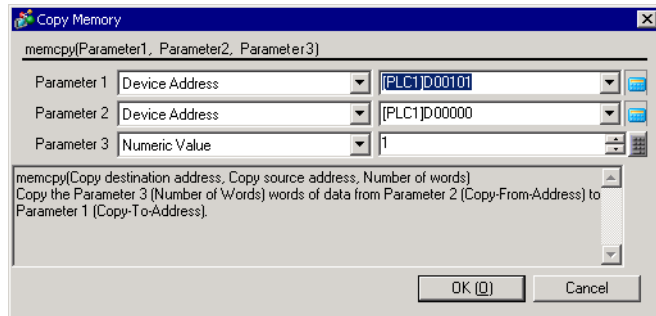
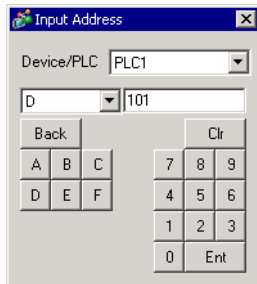
6 จาก [Built-in Function (Instruction)] ให้เลือก [Memory Operation]



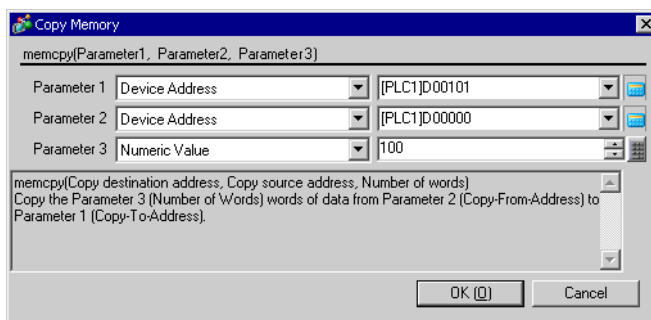
7 ดับเบิลคลิกที่ [Copy Memory] คลิก 



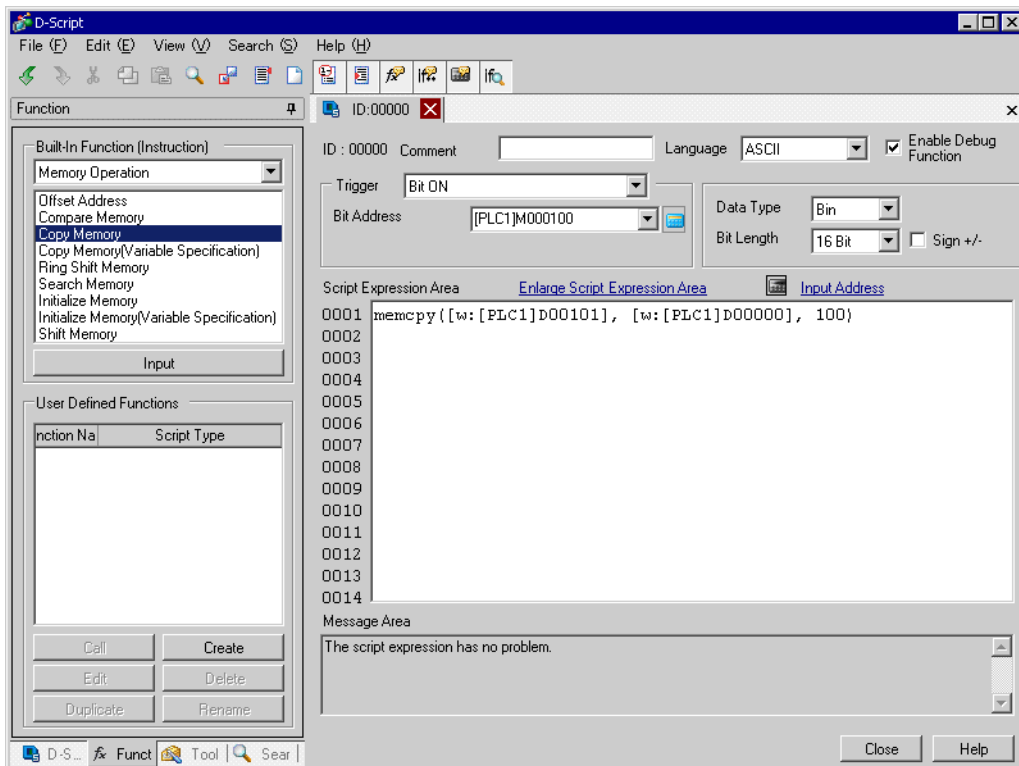
8 สำหรับ [Parameter 1] ให้ป้อน D00101 และคลิก [ENT]



9 สำหรับ [Parameter 2] ให้ป้อน D00000 และคลิก [OK]



10 สคริปต์นี้เสร็จสมบูรณ์แล้ว



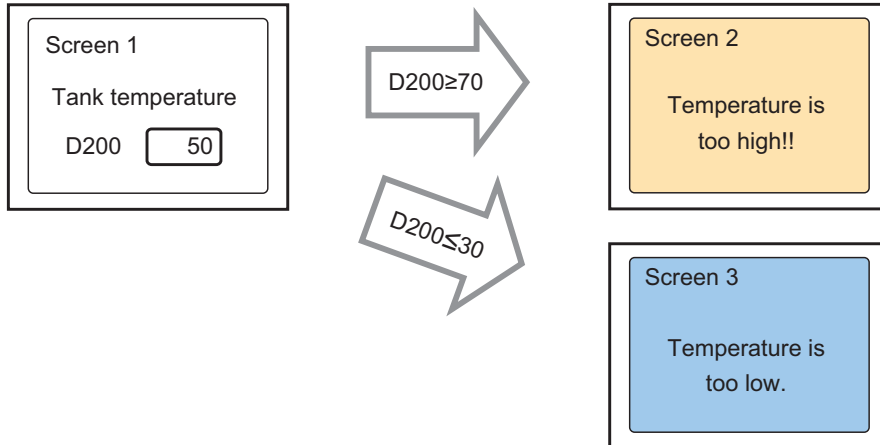
20.4 การแสดงการแจ้งเตือนเมื่อเกิดข้อผิดพลาด

หมายเหตุ

- โปรดอ่านรายละเอียดจากคำแนะนำในการตั้งค่า
☞ “20.8.1 คำแนะนำในการตั้งค่า D-Script/[Global D-Script] ทั่วไป” (หน้า 20-53)

การดำเนินการ

ระบบจัดการอุณหภูมิจะตรวจหาบิตที่มีข้อผิดพลาดจากอุปกรณ์ที่เชื่อมต่อไว้ และแสดงข้อความแจ้งเตือนเมื่อตำแหน่งจัดเก็บข้อมูลอุณหภูมิ (D200) มีค่าเพิ่มขึ้นถึงระดับตั้งแต่ 70 C ขึ้นไป หรือมีค่าลดลงถึงระดับตั้งแต่ 30 C ลงไป โดยสคริปต์นี้จะนับจำนวนข้อผิดพลาดที่ตรวจพบด้วย



ตำแหน่งที่ทำการนับทุกครั้งที่อยู่อุณหภูมิที่ D200 เพิ่มขึ้นถึงระดับตั้งแต่ 70 C ขึ้นไป และบันทึกจำนวนครั้งที่เกิดขึ้น: LS0300

ตำแหน่งที่ทำการนับทุกครั้งที่อยู่อุณหภูมิที่ D200 ลดลงถึงระดับตั้งแต่ 30 C ลงไป และบันทึกจำนวนครั้งที่เกิดขึ้น: LS0301

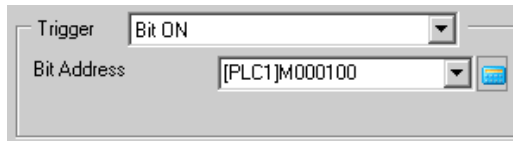
ตำแหน่งที่จัดเก็บหมายเลขหน้าจอเพื่อแสดงหน้าจอแจ้งเตือน:LS0302

คำสั่งที่ใช้

คำสั่ง	ข้อมูลสรุปของฟังก์ชัน
if ()	เมื่อเงื่อนไขภายในวงเล็บเป็นจริง ระบบจะดำเนินการกระบวนการที่อยู่หลังข้อความคำสั่ง “if ()”
greater than or equal to (>=)	เป็นจริงหาก N1 มากกว่าหรือเท่ากับ N2 (N1 >= N2)
Assignment (=)	กำหนดค่าฝั่งขวาให้แก่ค่าฝั่งซ้าย
Addition (+)	เพิ่มค่าคงที่ให้แก่ข้อมูลของอุปกรณ์ชนิดเวิร์ด
less than or equal to (<=)	เป็นจริงหาก N1 น้อยกว่าหรือเท่ากับ N2 (N1 <= N2)

เงื่อนไขการทริกเกอร์

ใน [Trigger Condition] ให้เลือก [Bit ON] และกำหนด [Bit Address] เป็น M000100




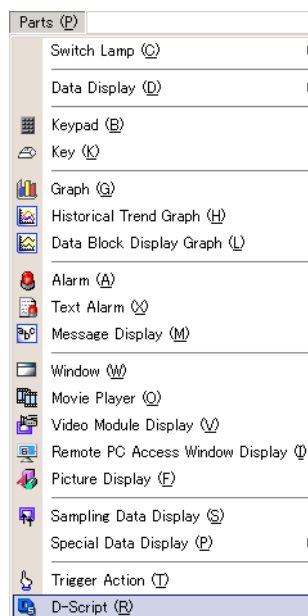
สคริปต์ที่เสร็จแล้ว

```

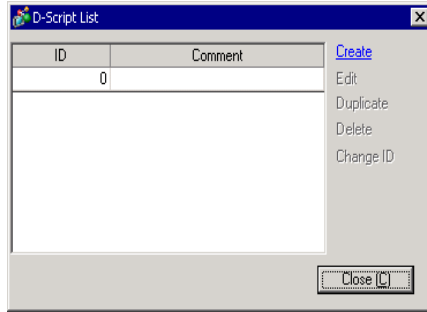
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 if ([w:[PLC1]D00200]>=70)           //When temp is greater than 70 degrees
0002 {
0003     [w:[#INTERNAL]LS0302]=100         //Greater than 70 degrees alarm screen number
0004     [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1 //Increase error count
0005 }
0006 endif
0007
0008 if ([w:[PLC1]D00200]>=30)           //When temp is greater than 30 degrees
0009 {
0010     [w:[#INTERNAL]LS0302]=101         //Greater than 30 degrees alarm screen number
0011     [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1 //Increase error count
0012 }
0013 endif
0014
    
```

ขั้นตอนการสร้าง

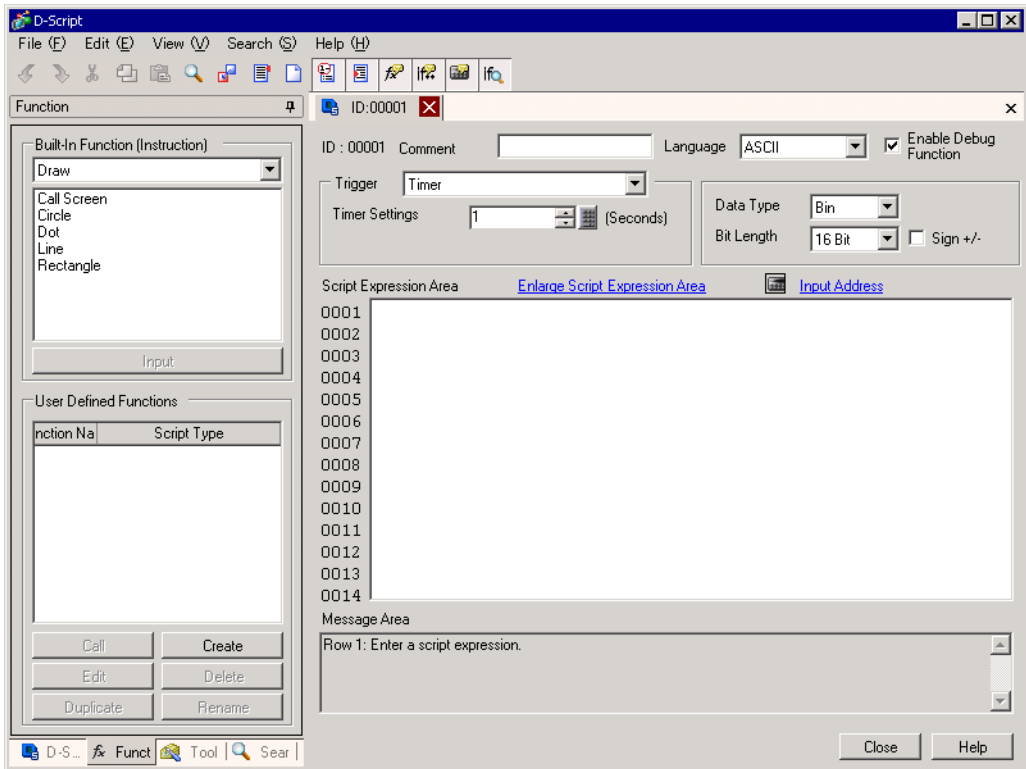
- 1 จากเมนู [Parts] ให้คลิกที่ [D-Script (R)] หรือคลิกที่ 



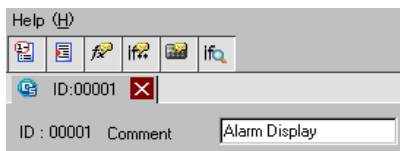
2 คลิก [Create] ID ของสคริปต์ที่มีอยู่จะแสดงอยู่ใน [D-Script List]



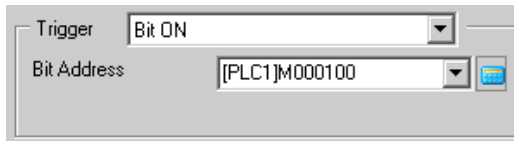
3 กล่องโต้ตอบ [D-Script] จะปรากฏขึ้น



4 ใน [Comment] ให้ป้อน “Alarm Display”



5 ใน [Trigger Conditions] ให้เลือก [Bit ON] และระบุ [Bit Address] เป็น M00100



6 สร้างโปรแกรมในพื้นที่ทำงาน โดยป้อนฟังก์ชัน ข้อความคำสั่ง และนิพจน์ การตั้งค่าเสร็จสมบูรณ์แล้ว

```

Script Expression Area      Enlarge Script Expression Area      Input Address
0001 if ([w:[PLC1]D00200]>=70)                //When temp is greater than 70 degrees
0002 {
0003     [w:[#INTERNAL]LS0302]=100                //Greater than 70 degrees alarm screen number
0004     [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1    //Increase error count
0005 }
0006 endif
0007
0008 if ([w:[PLC1]D00200]>=30)                //When temp is greater than 30 degrees
0009 {
0010     [w:[#INTERNAL]LS0302]=101                //Greater than 30 degrees alarm screen number
0011     [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1    //Increase error count
0012 }
0013 endif
0014
    
```

หมายเหตุ

- เมื่อเลือกข้อความ ให้กดปุ่ม [Ctrl] + [Shift] + [->] หรือ [-<] เพื่อเลือกข้อความทั้งบล็อก
- กดปุ่ม [Ctrl] + [F4] เพื่อปิดการแสดงผลที่เลือกในปัจจุบัน
- กดปุ่ม [Esc] เพื่อเขียนทับสคริปต์ หรือลบและออกจากการทำงาน

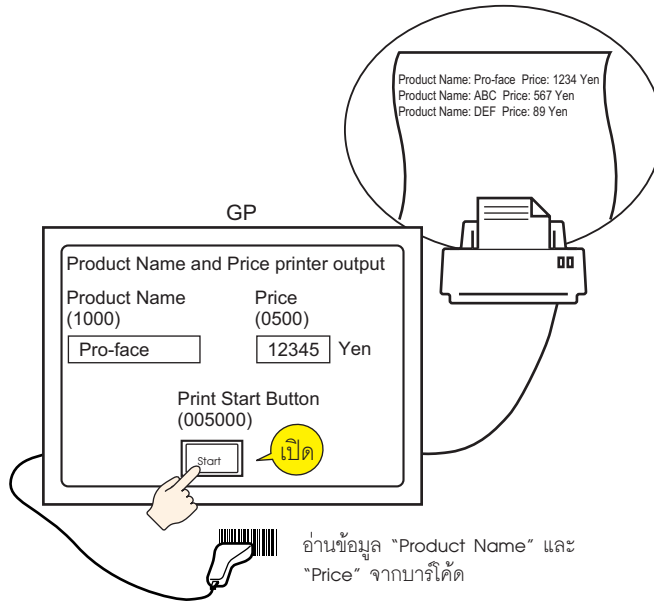
20.5 การสื่อสารกับอุปกรณ์ต่อพ่วงที่ไม่รองรับ

หมายเหตุ

- โปรดอ่านรายละเอียดจากคำแนะนำในการตั้งค่า
☞ “20.8.1 คำแนะนำในการตั้งค่า D-Script/[Global D-Script] ทั่วไป” (หน้า 20-53)

■ การดำเนินการ

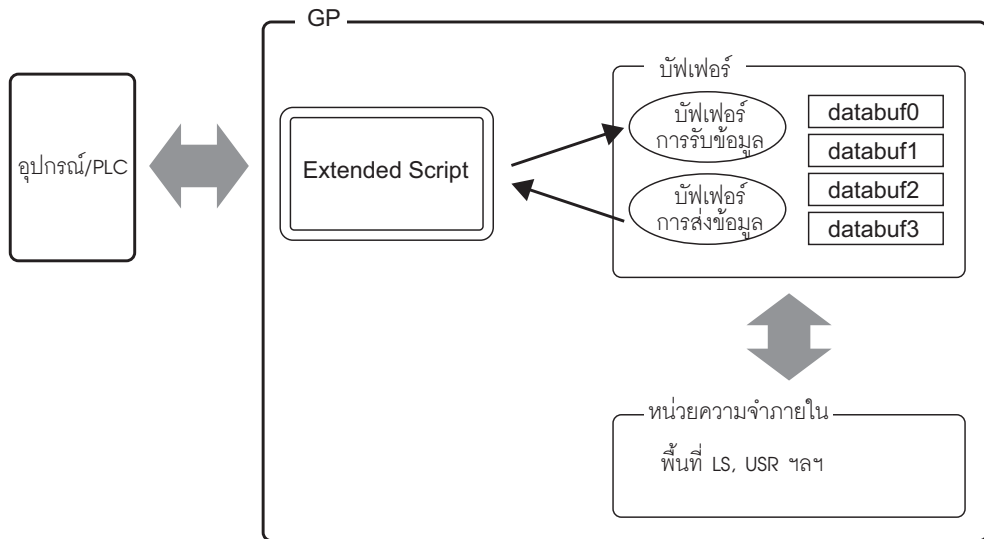
ตัวอย่างนี้จะเป็นการสร้าง Extended Script เพื่อส่งข้อมูลที่สามารถบาร์โค้ดซึ่งเชื่อมต่อกับพอร์ต USB ไปยังเครื่องพิมพ์ที่ต่อกับพอร์ต COM1 แบบอนุกรม



■ โครงสร้างของ Extended Script

Extended Script คือ สคริปต์ที่ใช้สื่อสารระหว่างพอร์ตอนุกรมภายในของ GP กับอุปกรณ์อินพุต/เอาต์พุตที่เชื่อมต่ออยู่

สำหรับการจัดการข้อมูล Extended Script ตามที่แสดงในรูปภาพด้านล่างนี้ ข้อมูลจะถูกเก็บไว้ใน databuf0 ถึง databuf3 โดยผ่านทางบัฟเฟอร์การส่ง/รับข้อมูล Databuf ไม่ได้แบ่งตามตำแหน่ง ดังนั้น โปรตเก็บข้อมูลไว้ในหน่วยความจำภายในก่อนจะแก้ไขข้อมูลในอุปกรณ์/PLC



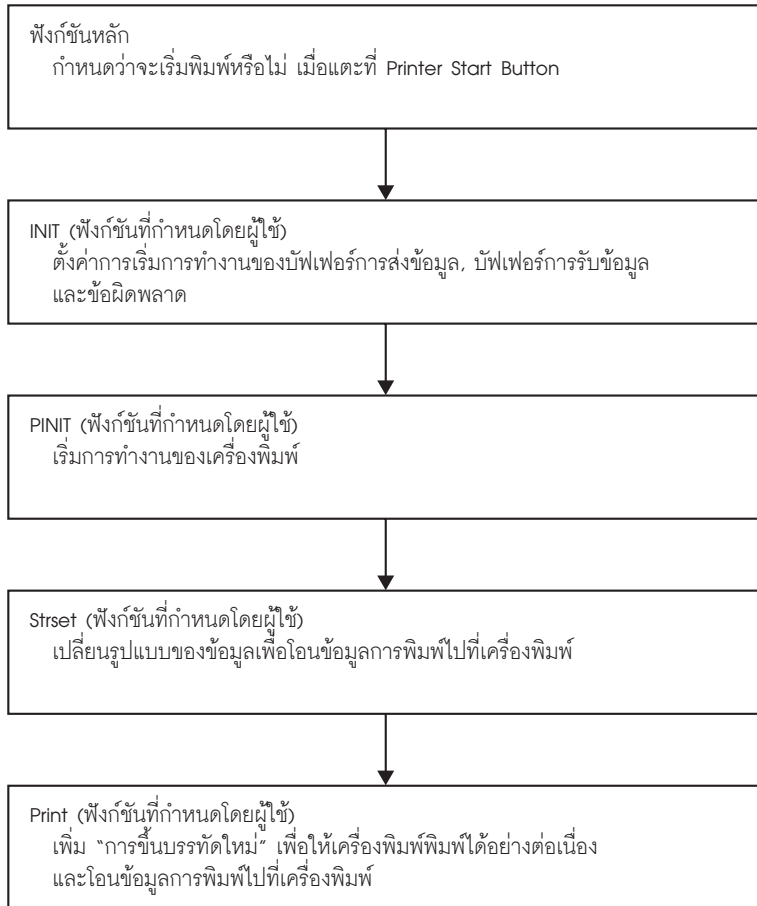
บัฟเฟอร์การรับข้อมูล/บัฟเฟอร์การส่งข้อมูล

สำหรับการสื่อสารกับอุปกรณ์/PLC บัฟเฟอร์นี้จะทำหน้าที่เป็นพื้นที่หน่วยความจำบิตซึ่งจะคอยแยกแยะข้อมูลที่ได้รับและข้อมูลที่ส่งในรูปแบบเรียลไทม์

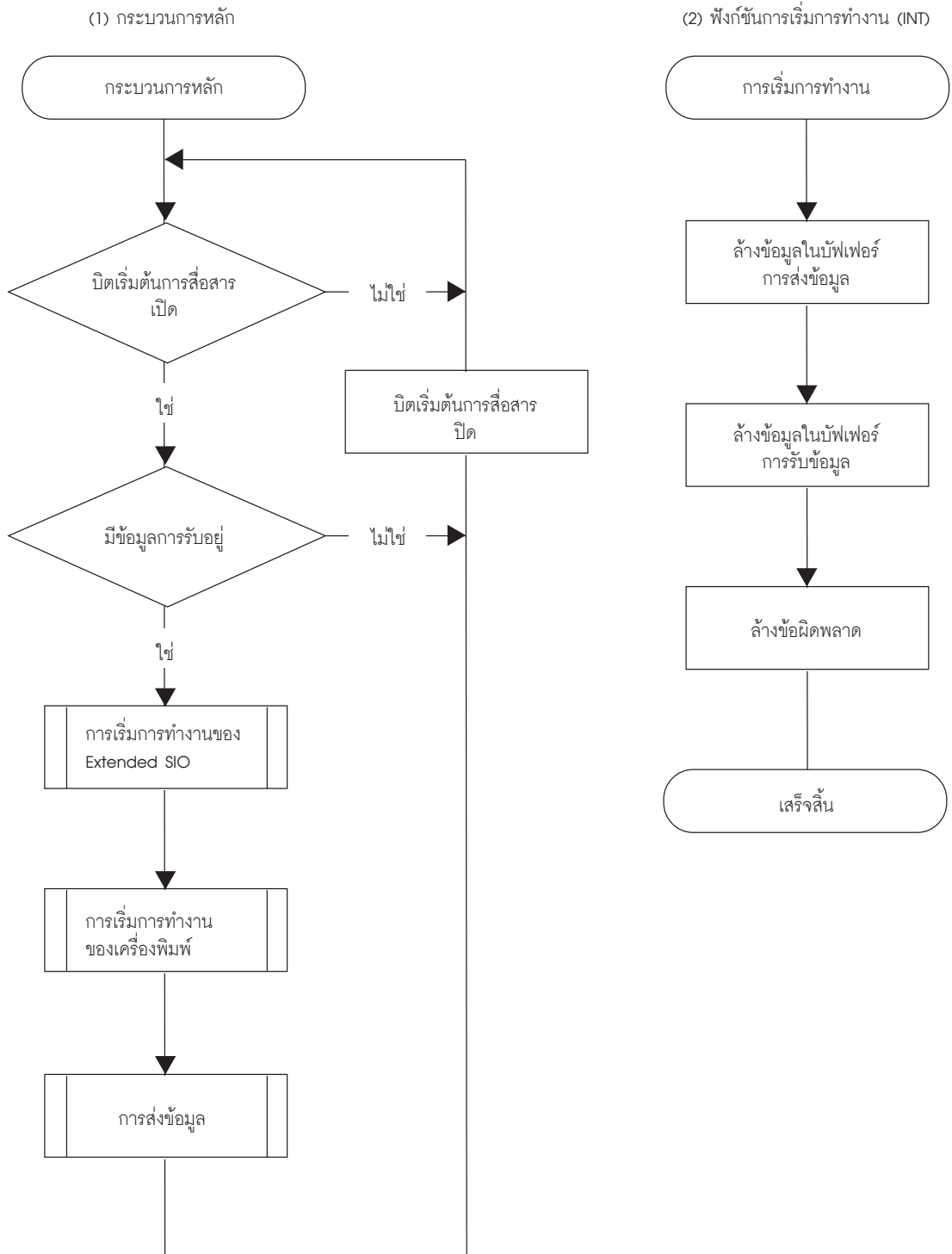
databuf0-databuf3

พื้นที่หน่วยความจำแบบไบต์ (8 บิต) ที่ใช้เก็บข้อมูล ขนาดของบัฟเฟอร์ คือ 1 KB

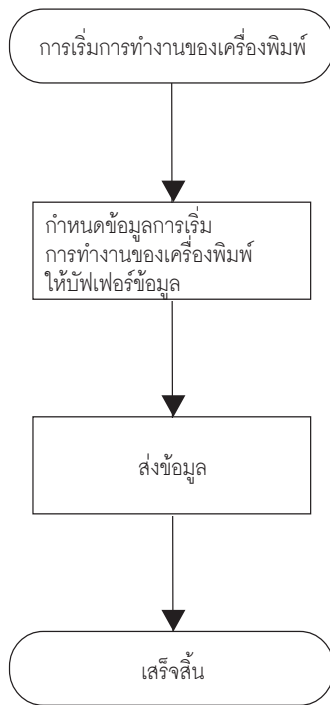
■ ขั้นตอนการสร้างสคริปต์



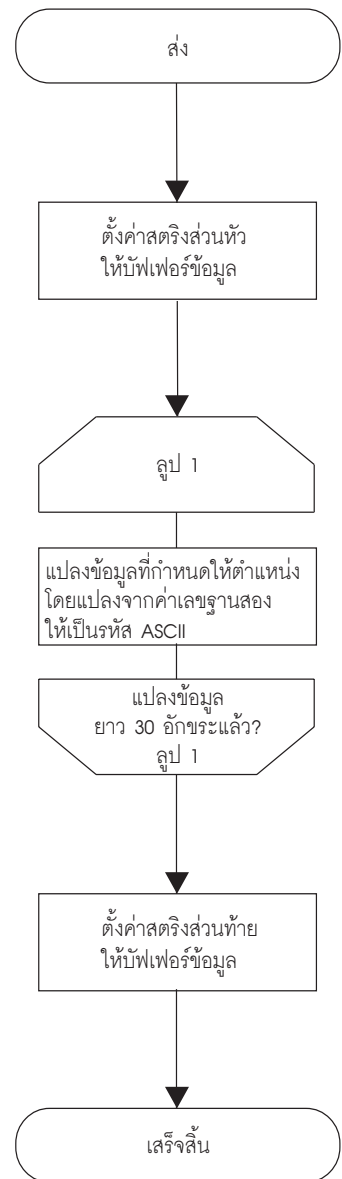
■ ผังการทำงาน



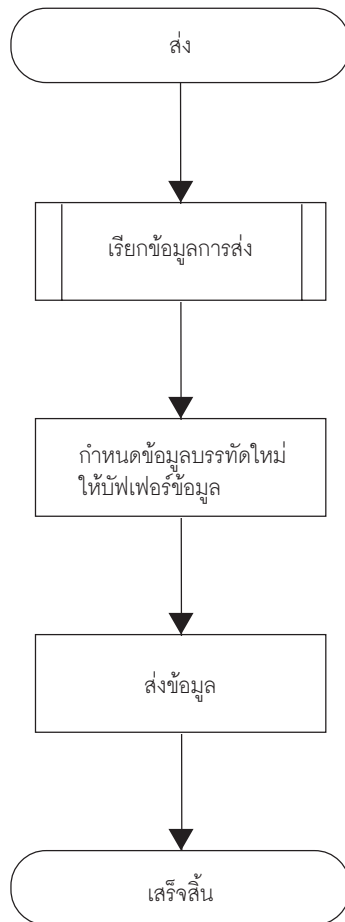
(3) ฟังก์ชันการเริ่มการทำงานของเครื่องพิมพ์ (PRINT)



(4) ฟังก์ชันสตริง (Strset)

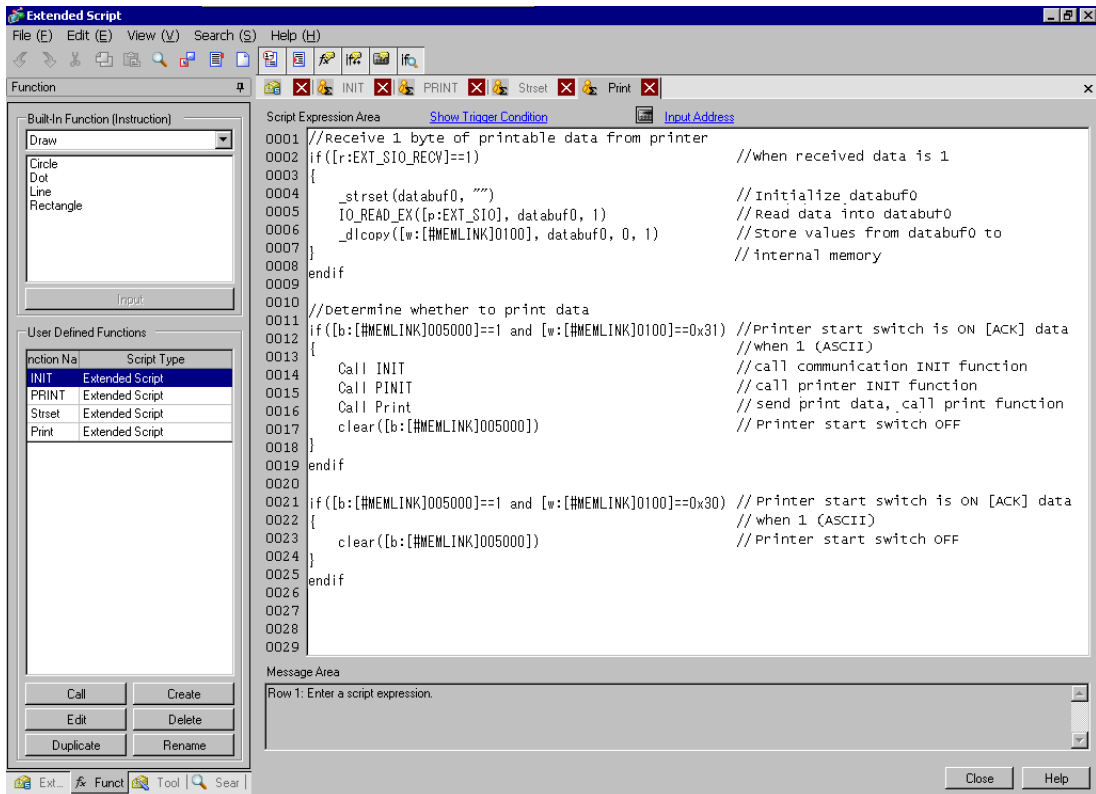


(5) ฟังก์ชันส่ง (Print)



■ สรุปฟังก์ชันของสคริปต์

- ◆ ฟังก์ชันหลัก
 - สคริปต์ที่เสร็จแล้ว



ข้อมูลสรุปของฟังก์ชัน

เมื่อปุ่ม Start ของเครื่องพิมพ์ (หน่วยความจำภายใน 005000) มีสถานะ ON ให้พิจารณาว่าจะเริ่มพิมพ์จากไบต์แรกของข้อมูล Print Permit หรือไม่

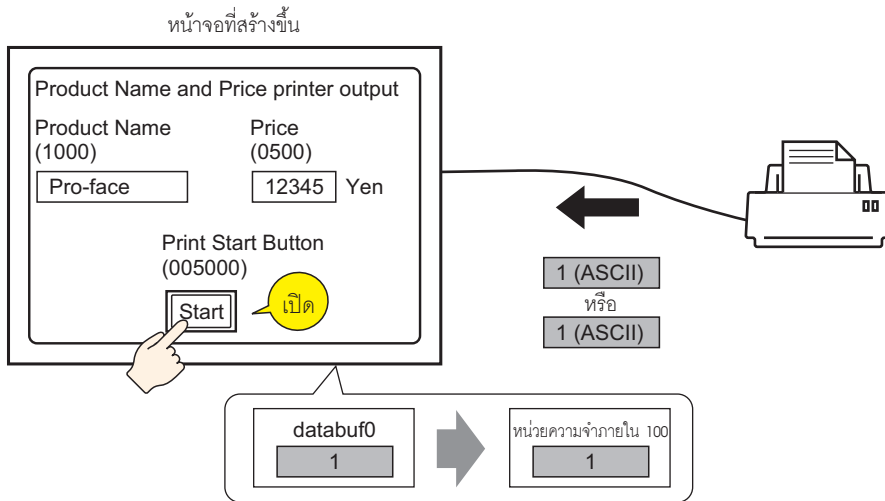
ข้อมูล Print Permit จะทำงานต่อไปนี้เป็นตัวอย่างข้อมูลจำเพาะของเครื่องพิมพ์

Print Preparation OK: Send 0x31 (ASCII code "1") to the device/PLC.

Print Preparation Invalid: Send 0x30 (ASCII code "0") to the device/PLC.

GP จะรับข้อมูล Print Permit ใน databuf0 แล้วย้ายข้อมูลนี้ไปที่หน่วยความจำภายใน 100 ซึ่งเข้าถึงได้ด้วยการจัดการสคริปต์ดังต่อไปนี้

เมื่อหน่วยความจำภายใน 100 ได้รับข้อมูล 0x31 (รหัส ASCII สำหรับค่า “1”) การพิมพ์จะเริ่มขึ้น เมื่อหน่วยความจำภายในได้รับข้อมูล 0x30 (รหัส ASCII สำหรับ “0”) GP จะกลับไปยังจุดเริ่มต้น แล้วปฏิบัติขั้นตอนนี้ซ้ำจนกว่าจะได้รับข้อมูล 0x31



◆ INIT (ฟังก์ชันที่กำหนดโดยผู้ใช้)

สคริปต์ที่เสร็จแล้ว


```
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 [c:EXT_SIO_CTRL00]=1      //Send buffer clear
0002 [c:EXT_SIO_CTRL01]=1      //Receive buffer clear
0003 [c:EXT_SIO_CTRL02]=1      //Error buffer clear
0004
```

ข้อมูลสรุปของฟังก์ชัน

ตั้งค่าการ Initialize ของบัพเฟอร์การส่งข้อมูล บัพเฟอร์การรับข้อมูล และข้อผิดพลาด

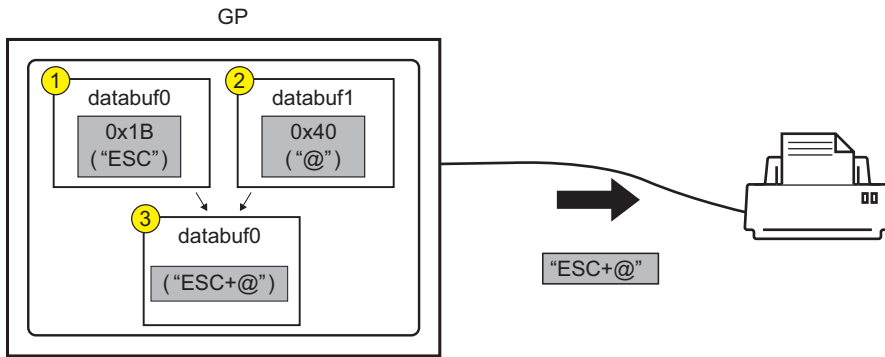
◆ PINIT (ฟังก์ชันที่กำหนดโดยผู้ใช้)

สคริปต์ที่เสร็จแล้ว

```
Script Expression Area Enlarge Script Expression Area  Input Address
0001 //Printer initialization(ESC/P command"ESC+@")
0002
0003 _strset(databuf0,"") //Clear databuf0
0004 _strset(databuf0, 0*1B) //Set ASCII code"ESC"
0005 _strset(databuf1,"") //Clear databuf1
0006 _strset(databuf1, 0*40) //Set ASCII code"@"
0007 _strset(databuf0,databuf1) //Add databuf1 to end of databuf0
0008 _strlen([t:0000],databuf0) //Store data length to temporary address
0009
0010 //Send data over serial port
0011
0012 IO_WRITE_EX([p:EXT_SIO],databuf0,[t:0000]) //Send databuf0,amount defined by temporary address value
0013
```

ข้อมูลสรุปของฟังก์ชัน

ทำการ Initialize เครื่องพิมพ์ ส่งคำสั่ง ESC/P “ESC+@” ไปยังเครื่องพิมพ์



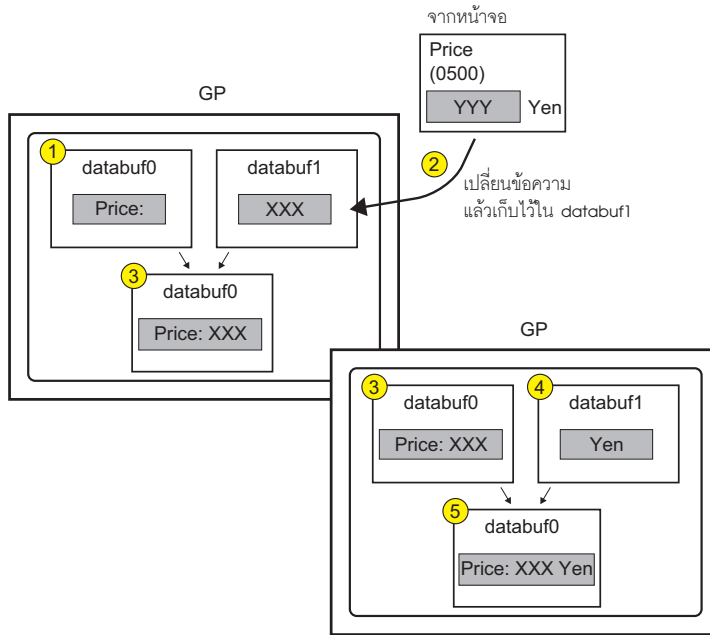
◆ Strset (ฟังก์ชันที่กำหนดโดยผู้ใช้)

สคริปต์ที่เสร็จแล้ว

Script	Expression Area	Enlarge Script Expression Area	Input Address
0001	//String example, add "Price:" and "\$"		
0002	_strset (databuf0, "")		//Initialize databuf0
0003	_strset (databuf0, "Price:")		//Store text "Price:" to databuf0
0004	bin2decase (databuf0, [w:[#MEMLINK]0500])		//Convert value to string and store in databuf1
0005	_strcat (databuf0, databuf1)		//Add databuf1 to end of databuf0
0006	_strset (databuf1, "")		//Initialize databuf1
0007	_strset (databuf1, "\$")		//Store text "\$" to databuf1
0008	_strcat (databuf0, databuf1)		//Add databuf1 to end of databuf0
0009			
0010	//Initialize temporary address		
0011	[t:0001]=0		
0012	[t:0002]=0		
0013			
0014	//Store to internal memory word units, consecutive characters into byte units(30 characters)		
0015	loop()		
0016	{		
0017	[w:[#MEMLINK]2000]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001]>>8		//Store top byte into bottom byte
0018	[w:[#MEMLINK]2001]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001]& 0xFF		//Erase top byte and store in next address
0019	[t:0001]=[t:0001]+1		//Address offset + 1
0020	[t:0002]=[t:0002]+2		//Address offset + 2
0021	if([t:0001]==15)		//Store 2 words into 2 byte and repeat 15 times
0022	{		
0023	break		
0024	}		
0025	endif		
0026	}		
0027	endloop		
0028	_ldcopy(databuf2, [w:[#MEMLINK]2000],30)		//Store internal memory 2000~2030 to data buffer as characters
0029			
0030	//Add string "Item:"		
0031	_strset (databuf1, "")		//Initialize databuf1
0032	_strset (databuf1, "Item:")		//Store string "Item:" into databuf1
0033	_strcat (databuf1, databuf2)		//Add databuf1 to end of databuf0
0034			
0035	//Add Item and Price strings		
0036	_strcat (databuf1, databuf0)		//Add databuf0 to end of databuf1

ข้อมูลสรุปของฟังก์ชัน

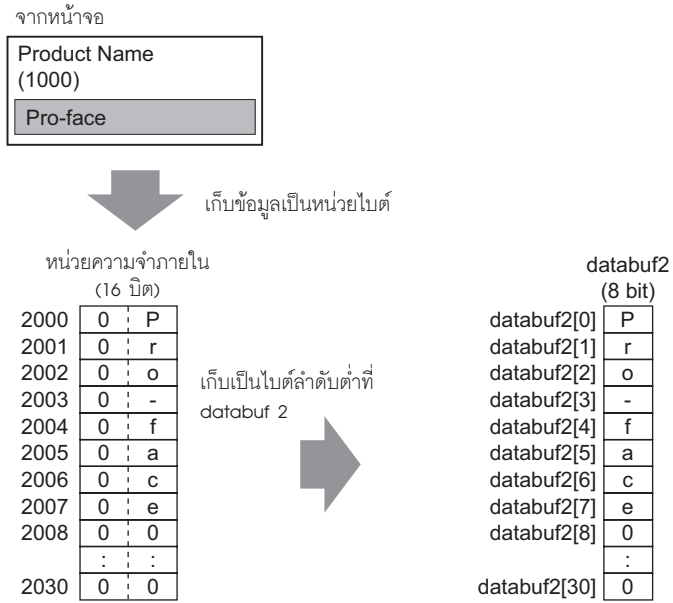
1 เพิ่มข้อความ “Price:” และ “Yen” ลงในข้อมูลราคาที่จัดเก็บอยู่ในตำแหน่ง 0500 ของหน่วยความจำภายใน



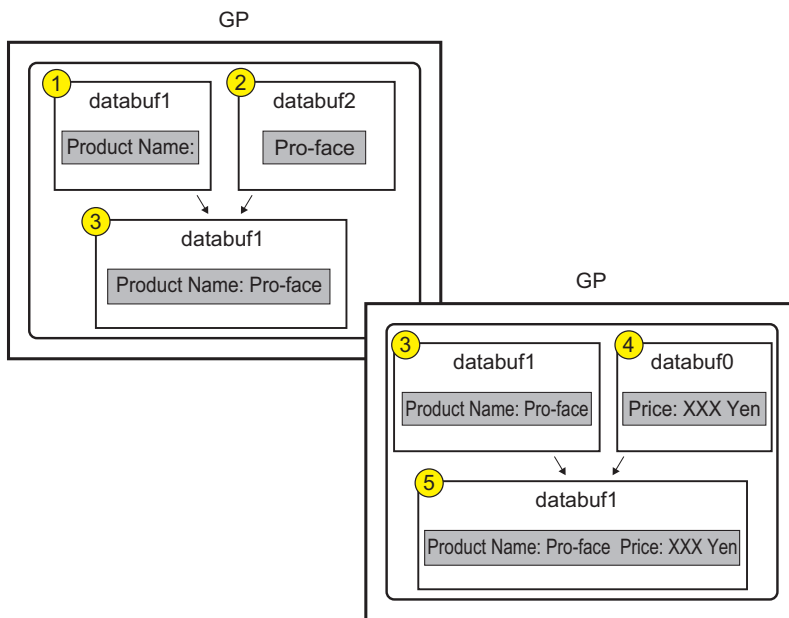
2 เปลี่ยนรูปแบบข้อมูลเพื่อส่งข้อมูลการพิมพ์ไปยังเครื่องพิมพ์ แบ่งข้อมูลสตริง (Product Name) ที่จัดเก็บไว้ตามลำดับในตำแหน่ง 1000 ของหน่วยความจำภายในออกเป็นหน่วยไบต์ แล้วเก็บข้อมูลดังกล่าวลงในตำแหน่ง 2000 ถึง 2030 ของหน่วยความจำภายใน โดยเก็บเป็นข้อมูลสตริงไบต์ล่าง ใช้ฟังก์ชัน _ldcopy และเก็บข้อมูลใน databuf2 ตามลำดับไบต์ล่างสุดของตำแหน่งเวิร์ดแบบต่อเนื่อง

หมายเหตุ

- ฟังก์ชัน _ldcopy จะนำข้อมูลไปเก็บไว้เป็นเวิร์ด และเก็บเฉพาะไบต์ลำดับต่ำกว่าในบัพเฟอร์ โดยจะไม่สนใจข้อมูลไบต์ที่มีลำดับสูงกว่า



3 เพิ่มข้อความ “Product Name:” และ “Price” ใน databuf2



◆ Print (ฟังก์ชันที่กำหนดโดยผู้ใช้)

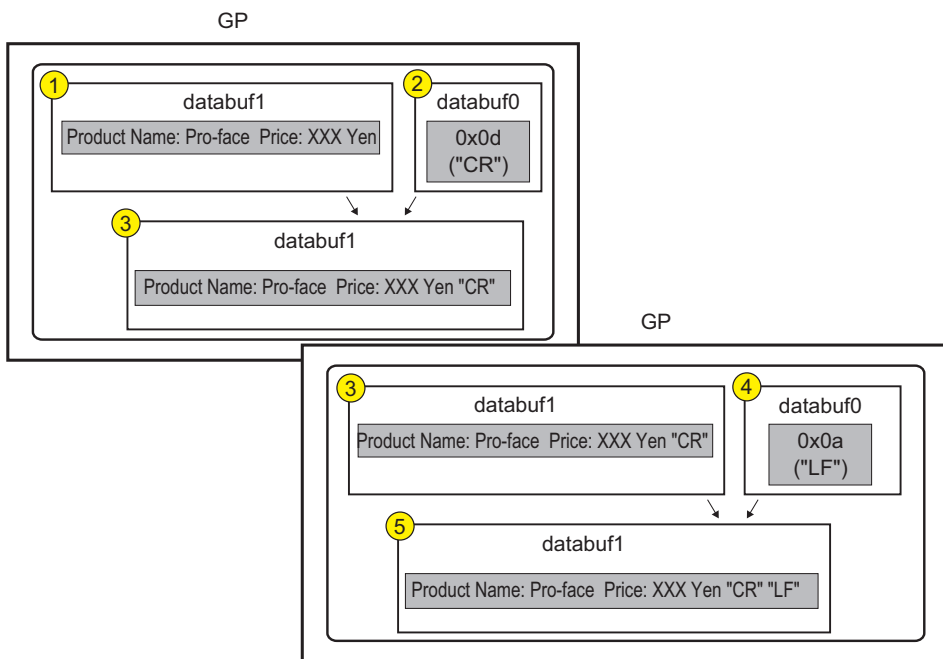
สคริปต์ที่เสร็จแล้ว

```

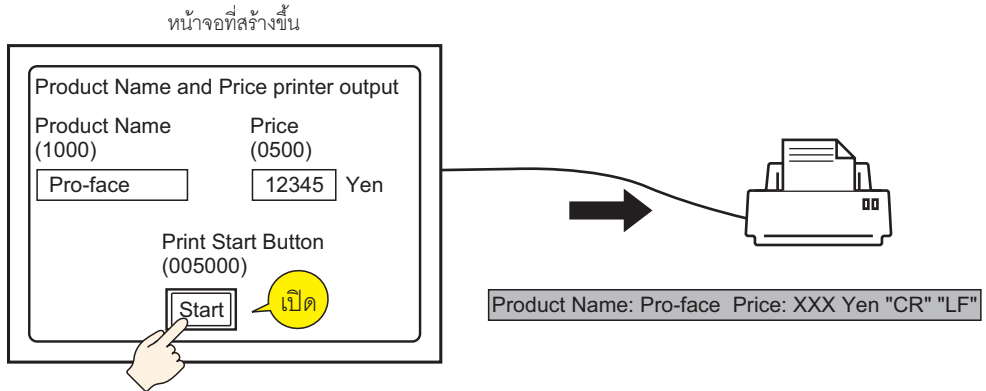
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 Call Street           //Call string data function
0002 _strset(databuf0,"")   //Clear databuf1
0003
0004 //Text delimiter
0005
0006 _strset(databuf0, 0*0d) //Return to start of row
0007 _strset(databuf1, databuf0) //Add databuf1 to end of databuf0
0008 _strset(databuf0, "") //Clear databuf1
0009 _strset(databuf0, 0*0a) //New line
0010 _strset(databuf1, databuf0) //Add databuf1 to end of databuf0
0011
0012 _strset([t:0000], databuf1) //Store data length to temporary address
0013
0014 //Send data over serial port
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000] //Send databuf0, amount defined by temporary address value
0017
    
```

ข้อมูลสรุปของฟังก์ชัน

1. เพิ่ม “การขึ้นบรรทัดใหม่” เพื่อให้เครื่องพิมพ์พิมพ์ได้อย่างต่อเนื่อง



2 ตั้งค่าข้อมูลการพิมพ์ให้เครื่องพิมพ์

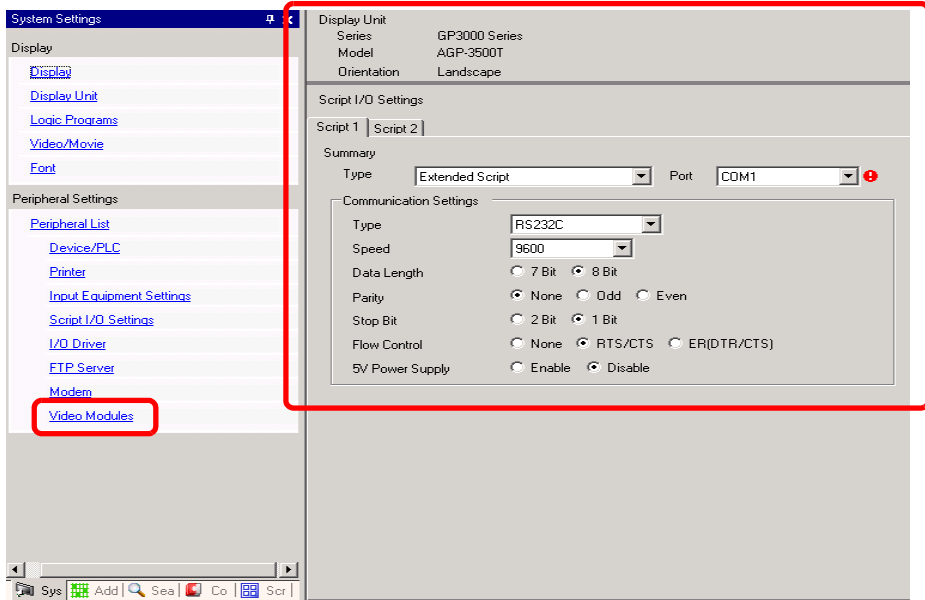


■ คำสั่งที่ใช้

คำสั่ง	ข้อมูลสรุปของฟังก์ชัน
if ()	เมื่อเงื่อนไขภายในวงเล็บเป็นจริง ระบบจะดำเนินการที่อยู่ที่หลังข้อความคำสั่ง "if ()"
Label Settings [r:EXT_SIO_RECV]	แสดงปริมาณข้อมูล (จำนวนไบต์) ที่ได้รับในคราวนั้นๆ ขนาดข้อมูลที่ได้รับจะมีคุณสมบัติเป็นแบบอ่านอย่างเดียว
Equivalent (==)	เป็นจริงหาก N1 เท่ากับ N2 (N1 = N2)
Text Settings (_strset)	จัดเก็บสตริงแบบตายตัวในบัพเฟอร์ข้อมูล
Extended Receive (IO_READ_EX)	รับข้อมูลจาก Extended SIO ตามขนาดที่ระบุในขนาดข้อมูลที่ได้รับ (ไบต์) แล้วเก็บไว้ในบัพเฟอร์ข้อมูล
From Data Buffer to Internal Device (_ldcopy)	ฟังก์ชันนี้จะคัดลอกข้อมูลสตริงแต่ละไบต์ที่เก็บไว้ในออฟเซตของบัพเฟอร์ข้อมูลไปยังพื้นที่ LS ตามจำนวนสตริง
Label Settings [c:EXT_SIO_CTRL **]	ตัวแปรควบคุมนี้ใช้สำหรับล้างข้อมูลในบัพเฟอร์การส่งข้อมูล บัพเฟอร์การรับข้อมูล และสถานะข้อผิดพลาด
Connect Text (_strcat)	เชื่อมสตริงอักขระหรือรหัสอักขระด้วยบัพเฟอร์ข้อความ
Text Length (_strlen)	รับความยาวของสตริงที่จัดเก็บไว้
Extended Send (IO_WRITE_EX)	ส่งข้อมูลในบัพเฟอร์ข้อมูลด้วย Extended SIO ตามขนาดจำนวนไบต์ที่ส่ง
Assignment (=)	กำหนดค่าฝั่งขวาให้แก่ค่าฝั่งซ้าย
Addition (+)	เพิ่มค่าคงที่ให้แก่ข้อมูลของอุปกรณ์ชนิดเวิร์ด
Numeric Value Decimal String Conversion (_bin2decasc)	ใช้ฟังก์ชันนี้เพื่อแปลงจำนวนเต็มให้เป็นสตริงเลขฐานสิบ
From Internal Device To Data Buffer (_ldcopy)	ข้อมูลของสตริงที่จัดเก็บอยู่ในพื้นที่ LS จะถูกคัดลอกไปยังบัพเฟอร์ข้อมูลตามจำนวนสตริงในการถ่ายโอนข้อมูลแบบไบต์ต่อไบต์

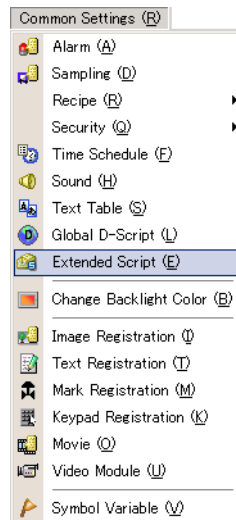
■ ขั้นตอนการสร้าง

- 1 ตั้งค่า Extended Script ที่ใช้สื่อสาร จากเมนู [Project (F)] ให้ชี้ที่ [System Settings (C)] และเลือก [Script Settings] แล้วตั้งค่า [Type] เป็น [Extended Script]

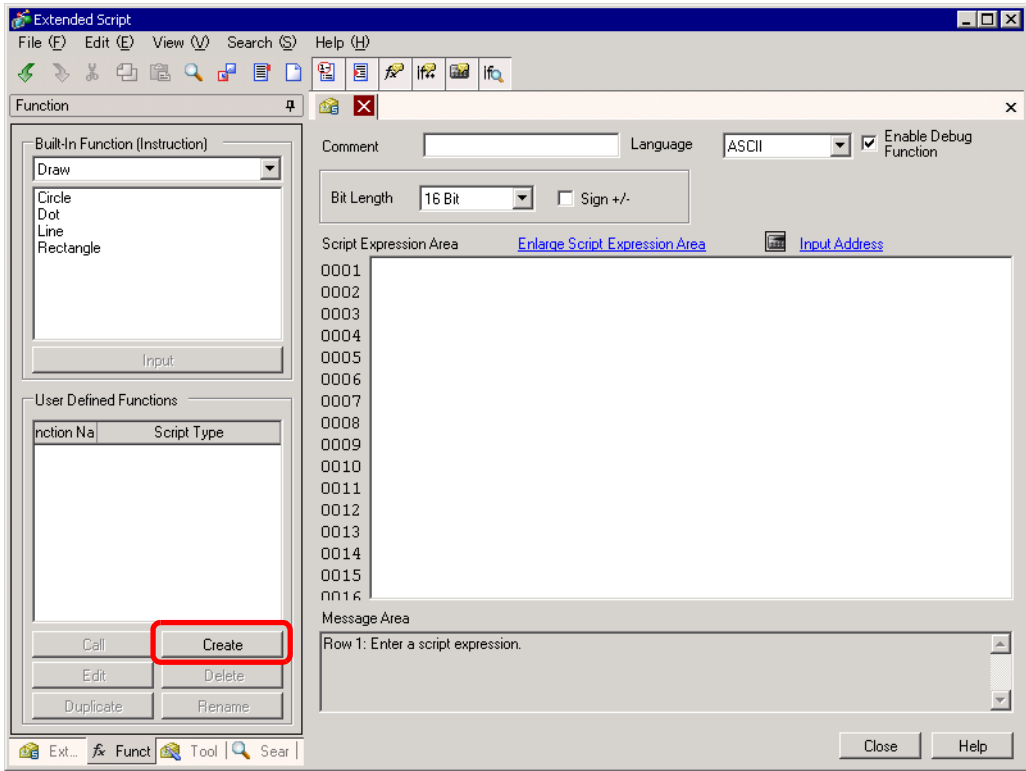


แท็บสำหรับตั้งค่าสคริปต์มีด้วยกัน 2 แท็บ

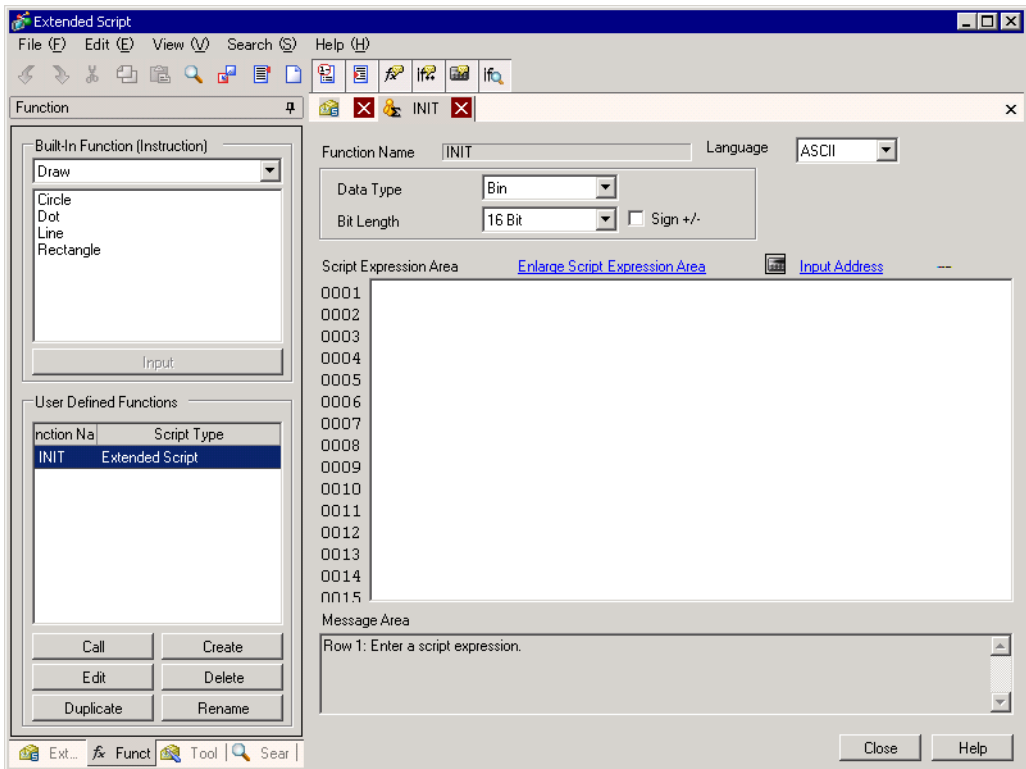
- 2 ตั้งค่า [Port] เป็น COM1 หรือ COM2 แล้วตั้งค่า [Communication Settings] ให้ตรงกับ Extended SIO จากเมนู [Common Settings (R)] ให้เลือก [Extended Script Settings (E)]



3 ลงทะเบียน “INIT” เป็นฟังก์ชันที่กำหนดโดยผู้ใช้ คลิกแท็บ [Function] แล้วคลิกปุ่ม [Create] ในกรอบ User Define Function



4 ป้อน [INIT] เป็นชื่อฟังก์ชัน แล้วคลิกที่ [OK] หน้าจอต่อไปนี้จะปรากฏขึ้น



5 สร้างสคริปต์ใน Execution Expression ด้วยการป้อนคำสั่ง ข้อความคำสั่ง และค่าคงที่

```
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 [c:EXT_SIO_CTRL00]=1      //Send buffer clear
0002 [c:EXT_SIO_CTRL01]=1      //Receive buffer clear
0003 [c:EXT_SIO_CTRL02]=1      //Error buffer clear
0004
```

6 ปฏิบัติตามขั้นตอนที่ผ่านมาเพื่อลงทะเบียน “PINIT” เป็นฟังก์ชันที่กำหนดโดยผู้ใช้ ป้อน [PINIT] เป็นชื่อฟังก์ชัน แล้วสร้างสคริปต์ต่อไปใน Execution Expression

```
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 //Printer initialization(ESC/P command"ESC+@")
0002
0003 _strset(databuf0,"")          //Clear databuf0
0004 _strset(databuf0, 0*1B)      //Set ASCII code"ESC"
0005 _strset(databuf1,"")        //Clear databuf1
0006 _strset(databuf1, 0*40)     //Set ASCII code"@"
0007 _strset(databuf0,databuf1)  //Add databuf1 to end of databuf0
0008 _strlen([t:0000],databuf0) //Store data length to temporary address
0009
0010 //Send data over serial port
0011
0012 IO_WRITE_EX([p:EXT_SIO],databuf0,[t:0000]) //Send databuf0,amount defined by temporary address value
0013
```

7 ปฏิบัติตามขั้นตอนที่ผ่านมาเพื่อลงทะเบียน “Strset” เป็นฟังก์ชันที่กำหนดโดยผู้ใช้ ป้อน [Strset] เป็นชื่อฟังก์ชัน แล้วสร้างสคริปต์ต่อไปใน Execution Expression

```
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 //String example, add "Price:" and "$"
0002 _strset(databuf0, "")          //Initialize databuf0
0003 _strset(databuf0, "Price:")    //Store text "Price:" to databuf0
0004 _bin2decasc(databuf0,[w:[#MEMLINK]0500]) //Convert value to string and store in databuf1
0005 _strcat(databuf0, databuf1)   //Add databuf1 to end of databuf0
0006 _strset(databuf1, "")        //Initialize databuf1
0007 _strset(databuf1, "$")       //Store text "$" to databuf1
0008 _strcat(databuf0, databuf1)  //Add databuf1 to end of databuf0
0009
0010 //Initialize temporary address
0011 [t:0001]=0
0012 [t:0002]=0
0013
0014 //Store to internal memory word units, consecutive characters into byte units(30 characters)
0015 loop()
0016 {
0017 [w:[#MEMLINK]2000][t:0002]=[w:[#MEMLINK]1000][t:0001]>>8 //Store top byte into bottom byte
0018 [w:[#MEMLINK]2001][t:0002]=[w:[#MEMLINK]1000][t:0001] & 0xFF //Erase top byte and store in next address
0019 [t:0001]=[t:0001]+1 //Address offset + 1
0020 [t:0002]=[t:0002]+2 //Address offset + 2
0021 if([t:0001]==15) //Store 2 words into 2 byte and repeat 15 times
0022 {
0023 break
0024 }
0025 endif
0026 }
0027 endloop
0028 _dcopy(databuf2, [w:[#MEMLINK]2000],30) //Store internal memory 2000~2030 to data buffer as characters
0029
0030 //Add string "Item:"
0031 _strset(databuf1, "")          //Initialize databuf1
0032 _strset(databuf1, "Item:")    //Store string "Item:" into databuf1
0033 _strcat(databuf1, databuf2)   //Add databuf1 to end of databuf0
0034
0035 //Add Item and Price strings
0036 _strcat(databuf1, databuf0)   //Add databuf0 to end of databuf1
```

(1)

(2)

(3)

8 ปฏิบัติตามขั้นตอนที่ผ่านมาเพื่อลงทะเบียน “Print” เป็นฟังก์ชันที่กำหนดโดยผู้ใช้ ป้อน [Print] เป็นชื่อฟังก์ชัน แล้วสร้างสคริปต์ต่อไปใน Execution Expression

```
Script Expression Area      Enlarge Script Expression Area  Input Address
0001 Call Street              //Call string data function
0002 _strset(databuf0,"")      //Clear databuf1
0003
0004 //Text delimiter
0005
0006 _strset(databuf0, 0*0d)    //Return to start of row
0007 _strset(databuf1, databuf0) //Add databuf1 to end of databuf0
0008 _strset(databuf0, "")    //Clear databuf1
0009 _strset(databuf0, 0*0a)  //New line
0010 _strset(databuf1, databuf0) //Add databuf1 to end of databuf0
0011
0012 _strset([t:0000], databuf1) //Store data length to temporary address
0013
0014 //Send data over serial port
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000] //Send databuf0, amount defined by temporary address valu
0017
```

9 สร้างสคริปต์หลัก สร้างสคริปต์ต่อไปใน Execution Expression การตั้งค่าจะเสร็จสมบูรณ์

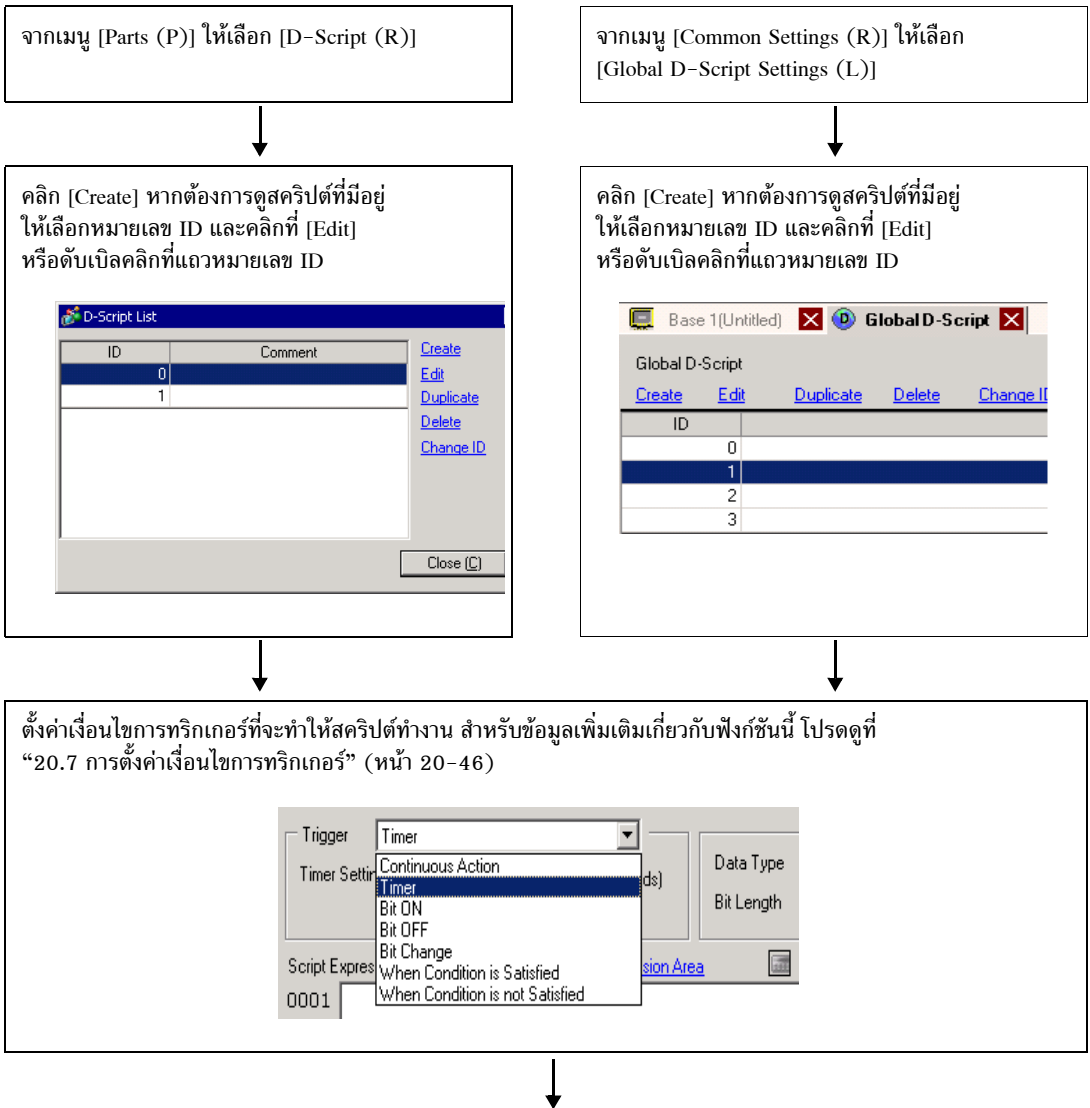
```
Script Expression Area      Enlarge Script Expression Area  Input Address
0001 //Receive 1 byte of printable data from printer
0002 if([r:EXT_SIO_RECV]==1) //When received data is 1
0003 {
0004     _strset(databuf0,"") //Initialize databuf0
0005     IO_READ_EX([p:EXT_SIO], databuf0, 1) //Read data into databuf0
0006     dlcoppy([w:[#MEMLINK]0100], databuf0, 0, 1) //Store values from databuf0 to internal
0007 }
0008 endif
0009
0010 //Determine whether to print data
0011 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0*31) //Printer start switch is ON [ACK] data
0012 {
0013     Call INIT //when 1 (ASCII)
0014     Call PINIT //call communication INIT function
0015     Call Print //call printer INIT function
0016     clear([b:[#MEMLINK]005000] //send print data, call print function
0017 //Printer start switch OFF
0018 }
0019 endif
0020
0021 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0*30) //Printer start switch is ON [ACK] data
0022 {
0023     clear([b:[#MEMLINK]005000] //when 0 (ASCII)
0024 //Printer start switch OFF
0025 }
0026 endif
```

หมายเหตุ

- เมื่อใส่ฟังก์ชันที่กำหนดโดยผู้ใช้ที่สร้างขึ้นในขั้นที่ 3 ถึงขั้นที่ 9 ลงในสคริปต์หลัก ให้เลือกฟังก์ชันที่จะใส่ แล้วคลิก [Call] ที่แท็บ [Function] ฟังก์ชันจะถูกใส่โดยใช้ “ชื่อฟังก์ชันที่เรียกใช้”
- เมื่อเลือกข้อความ ให้กดปุ่ม [Ctrl] + [Shift] + [->] หรือ [-<] เพื่อเลือกข้อความทั้งบรรทัด
- กดปุ่ม [Ctrl] + [F4] เพื่อปิดการแสดงผลที่เลือกในปัจจุบัน
- กดปุ่ม [Esc] เพื่อเขียนทับสคริปต์ หรือลบและออกจากการทำงาน

20.6 ขั้นตอนการสร้างสคริปต์

20.6.1 ขั้นตอนการสร้าง D-Scripts/Global D-Scripts

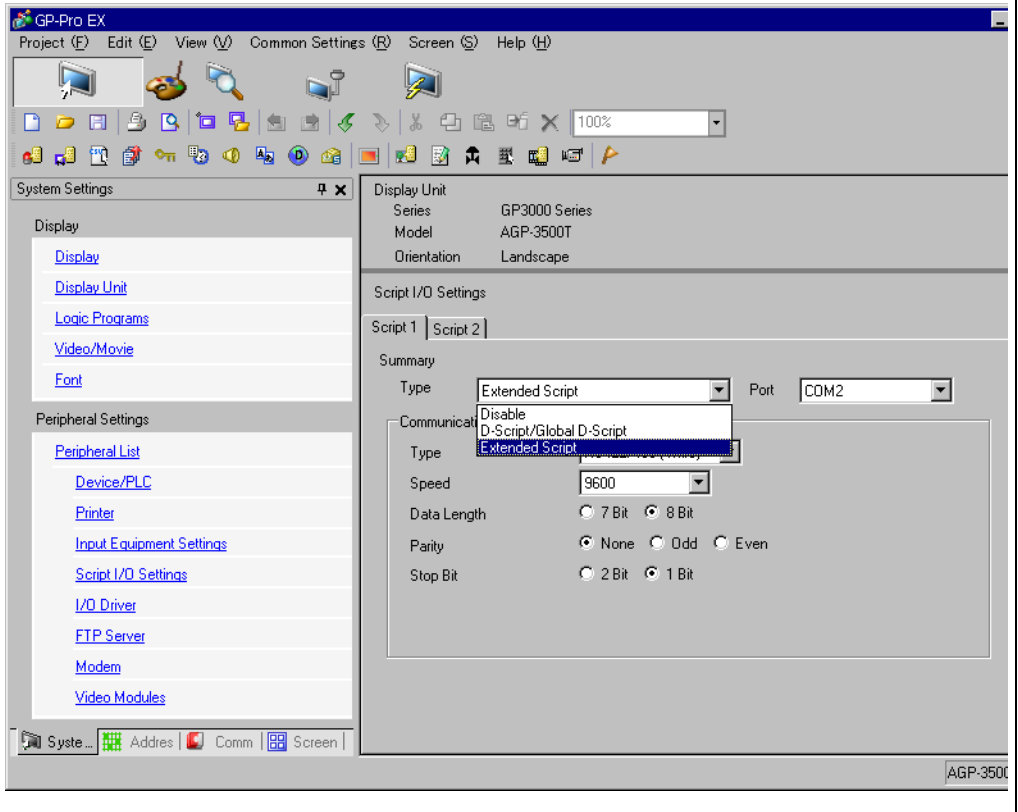


สร้างสคริปต์ (นิพจน์การดำเนินการ) หากต้องการทราบข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งและฟังก์ชันต่าง ๆ โปรดดูที่ “21.13 รายการคำสั่ง” (หน้า 21-100)

```
Script Expression Area      Enlarge Script Expression Area      Input Address
0001 [w:[PLC1]D00100]=[w:[PLC1]D00100]+1
0002 if ([w:[PLC1]D00100]==3)
0003 {
0004     [w:[#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
0008
0009
0010
0011
0012
0013
0014
```

20.6.2 ขั้นตอนการสร้าง Extended Script

จากเมนู [Project (F)] ให้เลือก [System Settings (C)] คลิกที่ [Script Settings] กล้องโต้ตอบดังต่อไปนี้จะปรากฏขึ้น หากใช้ Extended script ให้ตั้งค่า [Type] เป็น [Extended Script] และเลือก [Port]



จากเมนู [Common Settings (R)] ให้เลือก [Extended Script Settings (E)]



สร้างสคริปต์ (นิพจน์การดำเนินการ) หากต้องการทราบข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งและฟังก์ชันต่างๆ โปรดดูที่ “21 คำสั่งและคำอธิบาย” (หน้า 21-1)

```

Script Expression Area      Enlarge Script Expression Area      Input Address
0001 [พ: [PLC1]D00100]=[พ: [PLC1]D00100]+1
0002 if ([พ: [PLC1]D00100]==3)
0003 {
0004     [พ: [#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
0008
0009
0010
0011
0012
0013
0014
    
```

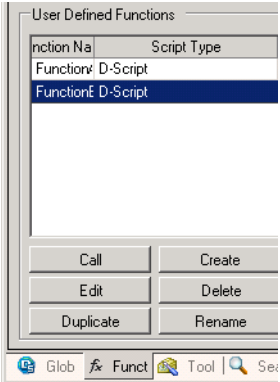
20.6.3 การตั้งค่าฟังก์ชันที่กำหนดโดยผู้ใช้

ลงทะเบียนสคริปต์ที่มีอยู่เป็นฟังก์ชันที่กำหนดโดยผู้ใช้ ดังนั้น คุณสามารถใช้สคริปต์เหล่านั้นในสคริปต์อื่น ๆ ได้ D-Script, Global D-Script หรือ Extended Script สามารถใช้ฟังก์ชันที่ลงทะเบียนไว้ได้

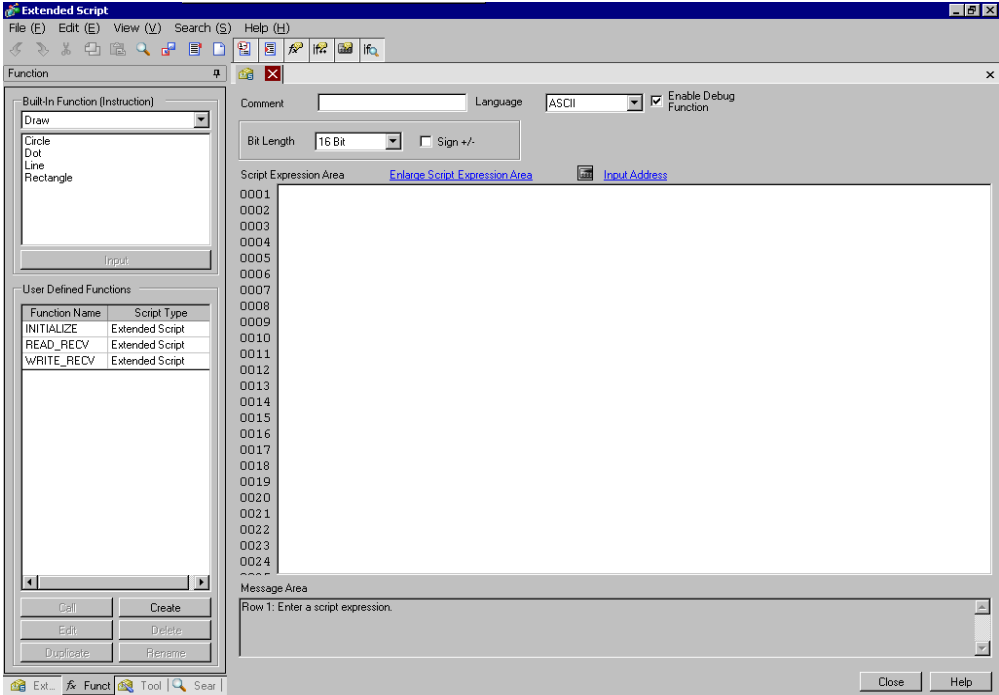
■ ขั้นตอนการตั้งค่า

เมื่อสร้างฟังก์ชันใหม่ที่กำหนดโดยผู้ใช้
คลิกที่ [Create] กล้องโต้ตอบ User-Defined Function จะปรากฏขึ้น

เมื่อแก้ไขฟังก์ชันที่กำหนดโดยผู้ใช้ที่ลงทะเบียนไว้
เลือกฟังก์ชันที่กำหนดโดยผู้ใช้ที่คุณต้องการแก้ไข แล้วคลิก [Edit] กล้องโต้ตอบ User-Defined Function จะปรากฏขึ้น



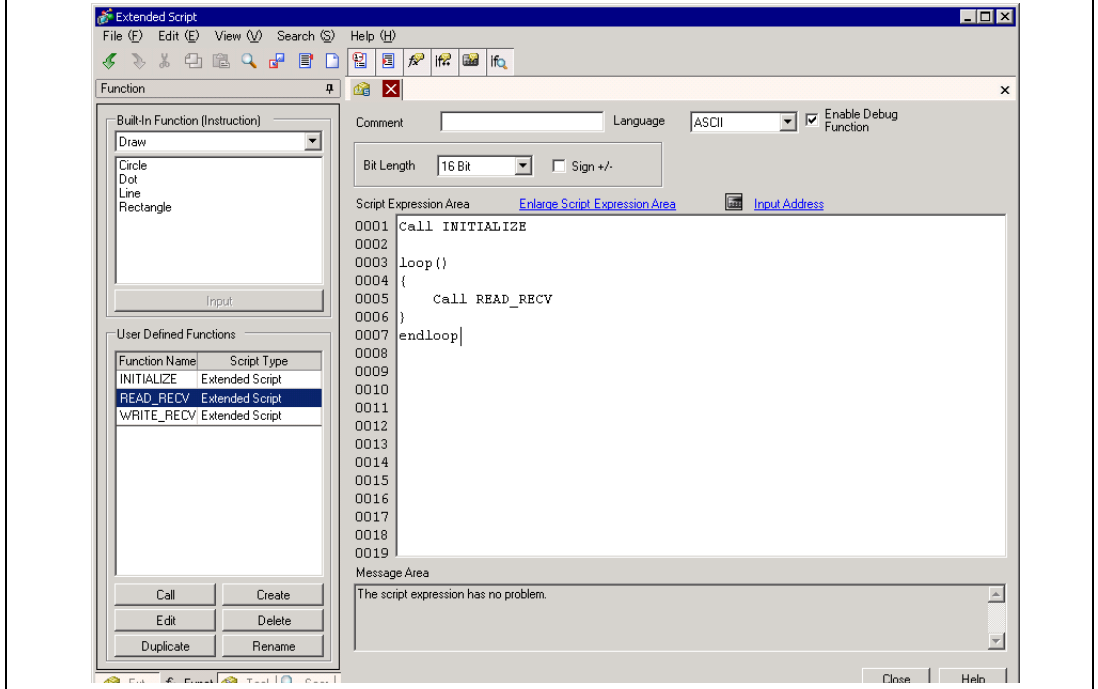
ป้อนชื่อฟังก์ชัน แล้วสร้างสคริปต์ในฟิลด์ Execution Expression คลิกที่ [OK] เพื่อบันทึกฟังก์ชันที่กำหนดโดยผู้ใช้



หมายเหตุ

- ข้อจำกัดใช้กับ Function Name โปรดดูข้อมูลเพิ่มเติมได้ที่ “20.9.3 ข้อจำกัดของฟังก์ชันที่กำหนดโดยผู้ใช้” (หน้า 20-65)

เลือกฟังก์ชันที่กำหนดโดยผู้ใช้ที่จะเรียกใช้งาน คลิก [Call] “ชื่อฟังก์ชันที่เรียกใช้” จะถูกใส่ลงในฟิลด์ Execution Expression



ข้อสำคัญ

- เมื่อฟังก์ชันที่กำหนดโดยผู้ใช้เรียกใช้งานสคริปต์อื่น ฟังก์ชันดังกล่าวจะไม่สามารถเรียกฟังก์ชันต่างๆ ที่สร้างขึ้นใน Extended Script ใน D-Scripts หรือ Global D-Scripts ได้

20.7 การตั้งค่าเงื่อนไขการทริกเกอร์

สคริปต์ที่สร้างขึ้นสามารถใช้เงื่อนไขการทริกเกอร์ 7 ชนิดดังต่อไปนี้

การตั้งค่า		คำอธิบาย
Continuous Action		สคริปต์ถูกทริกเกอร์ตามปกติ
ตัวตั้งเวลา		สคริปต์ถูกทริกเกอร์หลังจากครบระยะเวลาที่กำหนด
บิต	Bit ON	เมื่อ GP ตรวจพบบิตที่กำหนดมีค่าเพิ่มจาก 0 เป็น 1 สคริปต์จะถูกทริกเกอร์
	Bit OFF	เมื่อ GP ตรวจพบขอบขาลงของบิตที่กำหนด สคริปต์จะถูกทริกเกอร์
	Bit Change	เมื่อ GP ตรวจพบบิตที่กำหนดมีค่าเพิ่มจาก 0 เป็น 1 หรือลดลงจาก 1 เป็น 0 สคริปต์จะถูกทริกเกอร์
นิพจน์เงื่อนไข	เมื่อเงื่อนไขเป็นจริง	เมื่อ GP ตรวจพบว่านิพจน์ที่กำหนดไว้มีค่าเป็นจริง สคริปต์จะถูกทริกเกอร์
	เมื่อเงื่อนไขเป็นเท็จ	เมื่อ GP ตรวจพบว่านิพจน์ที่กำหนดไว้มีค่าเป็นเท็จ สคริปต์จะถูกทริกเกอร์

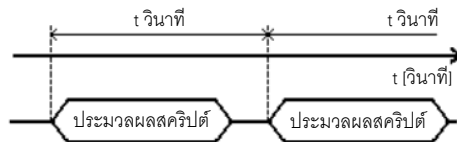
20.7.1 Continuous Action

ทำงานตามระยะเวลาสำหรับการแสดงผล

20.7.2 ตัวตั้งเวลา

■ ตัวตั้งเวลา

สคริปต์จะทำงานทุกครั้งที่ครบระยะเวลาที่กำหนดไว้ คุณสามารถตั้งระยะเวลาของตัวตั้งเวลานี้ได้ตั้งแต่ 1 ถึง 32,767 วินาที



หมายเหตุ

- ในการตั้งเวลาของฟังก์ชันตัวตั้งเวลา ค่าของเวลาจะประกอบด้วยเวลาที่กำหนดไว้ + ข้อผิดพลาดของเวลาสำหรับการแสดงผล นอกจากนี้ ฟังก์ชันตัวตั้งเวลาอาจทำงานได้ซ้ำขึ้นอยู่กับเวลาที่ช้ากว่ารายการหน้าจหรือเวลาที่ใช้พิมพ์ข้อมูลด้วย สำหรับข้อมูลเพิ่มเติมเกี่ยวกับเวลาสำหรับการแสดงผล โปรดดูที่ “ ■ ข้อจำกัดของทริกเกอร์บิต” (หน้า 20-50)
- เมื่อใช้ D-Script การเปลี่ยนหน้าจจะทำให้ฟังก์ชันตัวตั้งเวลาเริ่มนับจาก 0 ใหม่

20.7.3 บิต

■ Bit ON

เมื่อ GP ตรวจพบว่าตำแหน่งบิตที่กำหนดไว้ (ทริกเกอร์บิต) มีค่าเพิ่มจาก 0 เป็น 1 สคริปต์จะถูกทริกเกอร์



หมายเหตุ

- โปรดกำหนดระยะเวลาในการเปิดหรือปิดทริกเกอร์บิตให้มีระยะเวลานานกว่าระยะเวลาของรอบการสื่อสารหรือเวลาสำหรับการแสดงผล ทั้งนี้ขึ้นอยู่กับระยะเวลาใดนานกว่ากัน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฟังก์ชันนี้ โปรดดูที่ “ ■ ข้อจำกัดของทริกเกอร์บิต” (หน้า 20-50)

■ Bit OFF

เมื่อ GP ตรวจพบว่าตำแหน่งบิตที่กำหนดไว้ (ทริกเกอร์บิต) มีค่าลดลงจาก 1 เป็น 0 สคริปต์จะถูกทริกเกอร์



หมายเหตุ

- โปรดกำหนดระยะเวลาในการเปิดหรือปิดทริกเกอร์บิตให้มีระยะเวลานานกว่าระยะเวลาของรอบการสื่อสารหรือเวลาสำหรับการแสดงผล ทั้งนี้ขึ้นอยู่กับระยะเวลาใดนานกว่ากัน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฟังก์ชันนี้ โปรดดูที่ “ ■ ข้อจำกัดของทริกเกอร์บิต” (หน้า 20-50)

■ Bit Change

เมื่อ GP ตรวจพบว่าตำแหน่งบิตที่กำหนดไว้ (ทริกเกอร์บิต) มีค่าเพิ่มจาก 0 เป็น 1 หรือลดลงจาก 1 เป็น 0 สคริปต์จะถูกทริกเกอร์



หมายเหตุ

- โปรดกำหนดระยะเวลาในการเปิดหรือปิดทริกเกอร์บิตให้มีระยะเวลานานกว่าระยะเวลาของรอบการสื่อสารหรือเวลาสำหรับการแสดงผล ทั้งนี้ขึ้นอยู่กับระยะเวลาใดนานกว่ากัน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฟังก์ชันนี้ โปรดดูที่ “ ■ ข้อจำกัดของทริกเกอร์บิต” (หน้า 20-50)

20.7.4 นิพจน์เงื่อนไข

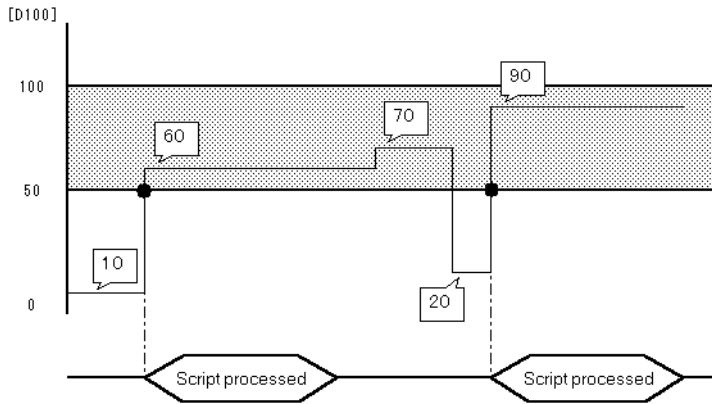
■ เมื่อเงื่อนไขเป็นจริง

เมื่อ GP ประเมินว่าเงื่อนไขการทริกเกอร์เป็นจริง สคริปต์จะเริ่มทำงาน

ตัวอย่างเช่น เมื่อเงื่อนไขการทริกเกอร์เป็น $100 > [D100] > 50$ สคริปต์จะทำงานตามระยะเวลาต่อไปนี้

เมื่อตรวจพบเงื่อนไขเปลี่ยนจาก [False]→[True] สคริปต์จะทำงาน และระบบจะระบุ 70 ให้กับตำแหน่ง D100

หากเงื่อนไขเปลี่ยนแปลงแบบ [True]→[True] สคริปต์จะไม่ทำงาน

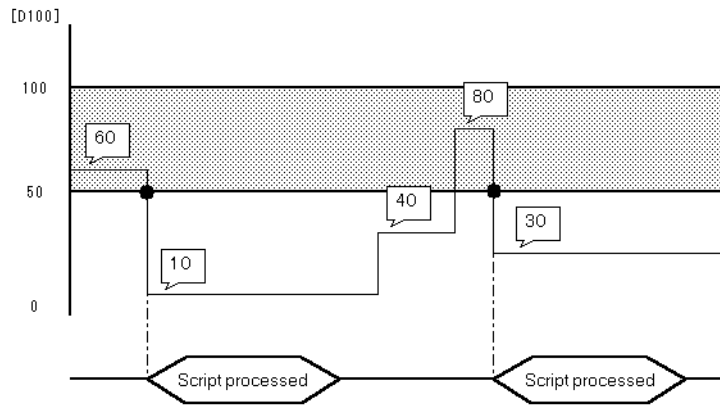


หมายเหตุ

- โปรดกำหนดระยะเวลาของเงื่อนไขการทริกเกอร์ให้มีระยะเวลานานกว่าระยะเวลาของรอบการสื่อสารหรือเวลาสำหรับการแสดงผล ขึ้นกับว่าระยะเวลาใดนานกว่ากัน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฟังก์ชันนี้ โปรดดูที่ “ ■ ข้อจำกัดของทริกเกอร์บิต” (หน้า 20-50)

■ เมื่อไม่ตรงตามเงื่อนไข

สคริปต์จะทำงานหนึ่งครั้งเมื่อ GP ตรวจสอบว่าฟังก์ชันที่กำหนดในโปรแกรมทริกเกอร์มีค่าเป็นเท็จ เมื่อเงื่อนไขการทริกเกอร์เป็น $100 > [D100] > 50$ สคริปต์จะทำงานตามระยะเวลาต่อไปนี้ เมื่อตรวจพบว่าเงื่อนไขเปลี่ยนจาก [True]→[False] สคริปต์จะทำงาน และกำหนดค่า 20 ให้ตำแหน่ง D100 แต่สคริปต์จะไม่ทำงานหากเงื่อนไขเปลี่ยนแปลงแบบ [False]→[False]



หมายเหตุ

- โปรดกำหนดระยะเวลาของเงื่อนไขการทริกเกอร์ให้มีระยะเวลานานกว่าระยะเวลาของรอบการสื่อสารหรือเวลาสำหรับการแสดงผล ขึ้นกับว่าระยะเวลาใดนานกว่ากัน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฟังก์ชันนี้ โปรดดูที่ “ ■ ข้อจำกัดของทริกเกอร์บิต” (หน้า 20-50)

■ ข้อจำกัดของทริกเกอร์บิต

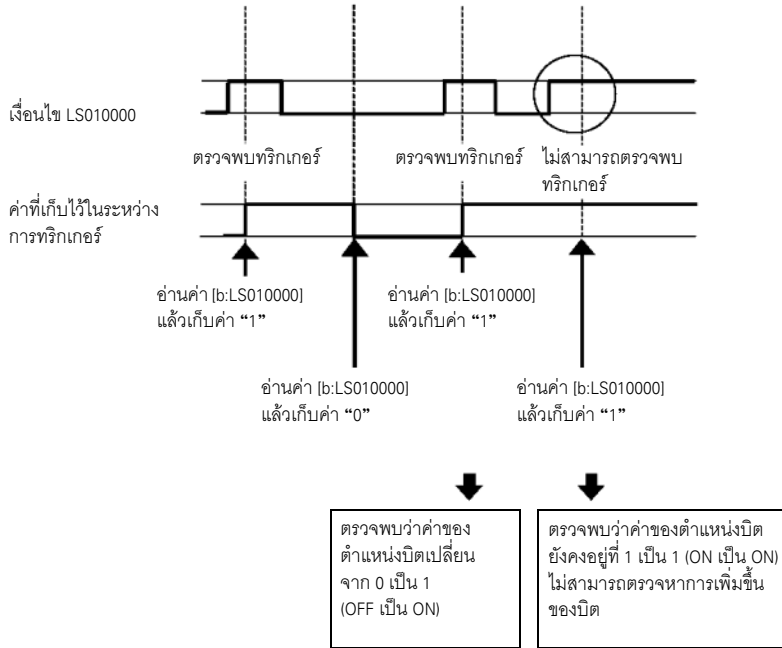
- โปรดกำหนดระยะเวลาในการเขียนข้อมูลลงในอุปกรณ์ที่เชื่อมต่อให้มีระยะเวลานานกว่าระยะเวลาของรอบการสื่อสาร การเขียนข้อมูลลงในอุปกรณ์ที่เชื่อมต่ออยู่โดยใช้ตัวนับจำนวนการสแกนของรีเลย์พิเศษภายในของ GP เป็นจำนวนบ่อยครั้ง อาจเกิดข้อผิดพลาดในการสื่อสารหรือข้อผิดพลาดของระบบได้
- เมื่อตั้งค่าบิตที่ใช้ในเงื่อนไขการทริกเกอร์ของ D-Script ให้เป็นการ “แตะ” แต่บิตนั้นปิดลงในขณะประมวลผล D-Script ระยะเวลาที่ใช้ในการแตะบริเวณสัมผัสซ้ำ ๆ หลายครั้งอาจทำให้ระบบตรวจไม่พบการเพิ่มขึ้นของบิตได้ ทริกเกอร์ D-Script จะเปรียบเทียบค่าที่อ่านในครั้งก่อนกับค่าที่อ่านได้ในปัจจุบัน เพื่อพิจารณาว่าทริกเกอร์ในขณะนี้มีค่าเป็น “จริง” หรือไม่ อย่างไรก็ตาม ในระหว่างการสแกนแต่ละครั้ง ค่าที่เก็บไว้ในตำแหน่งบิตที่ใช้ระหว่างการทริกเกอร์จะเป็นค่าเดียวกัน แม้ว่าค่านั้นจะเปลี่ยนไปในระหว่างที่สคริปต์ทำงานก็ตาม ระบบจะอ่านค่าใหม่หลังจากสแกนครั้งต่อไปเท่านั้น

ระยะเวลาของรอบการสื่อสาร ระยะเวลาของรอบการสื่อสาร คือเวลาที่ใช้เพื่อร้องขอและนำข้อมูลจาก GP ไปที่ PLC ข้อมูลนี้จะถูกจัดเก็บเป็นข้อมูลเลขฐานสองในตำแหน่ง LS2037 ของอุปกรณ์ภายใน โดยมีหน่วยเป็นมิลลิวินาที (ms) และมีความคลาดเคลื่อนเท่ากับ ± 10 มิลลิวินาที

เวลาสำหรับการแสดงผล: เวลาสำหรับการแสดงผล คือเวลาที่ใช้ในการแสดงผล/คำนวณค่าของหน้าจอ 1 หน้าจอ ข้อมูลนี้จะถูกจัดเก็บเป็นข้อมูลเลขฐานสองในตำแหน่ง LS2036 ของอุปกรณ์ภายใน โดยมีหน่วยเป็นมิลลิวินาที (ms) และมีความคลาดเคลื่อนเท่ากับ ± 10 มิลลิวินาที

ตัวอย่าง เมื่อกำหนดให้เปิดทริกเกอร์บิต (LS010000) ด้วยการแตะและปิดโดยใช้ D-Script
Triggered Condition: Bit ON [#INTERNAL] LS010000
Execution Expression: clear ([b:[#INTERNAL] LS010000])

◆ ผังเวลาการประมวลผล D-Script



ตามตัวอย่างนี้ หากไม่ใช้ระยะเวลาในการแตะของ D-Script และทำการตรวจหาเพียงอย่างเดียวเท่านั้น การประมวลผลจะเป็นดังนี้

การใช้ข้อความคำสั่ง if () เพื่อตรวจหาทริกเกอร์

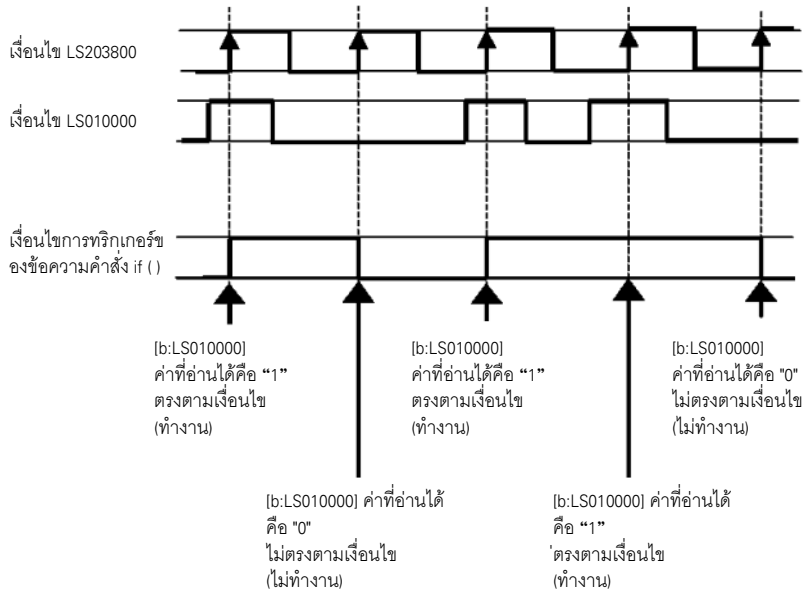
ใช้ข้อความคำสั่ง if เพื่อระบุว่าการแตะจะเป็นการกำหนดบิตหรือไม่ ทุกครั้งที่ใช้ข้อความคำสั่ง if () ระบบจะอ่านค่าและทำการเปรียบเทียบ

```
Triggered Condition: Bit ON ([#INTERNAL]LS203800 *1)
Execution Expression: if ([b:[#INTERNAL]LS010000]==1)
{
clear ([b:[#INTERNAL]LS010000])
:
:
}
```

ตัวนับภายในของ GP ตัวนับจะเพิ่มขึ้นทุกครั้งที่พาร์ทซึ่งตั้งค่าบนหน้าจอแสดงผลทำงาน

เมื่อสร้าง D-Script ชนิดนี้ ถึงแม้จะป้อนข้อมูลด้วยการแตะซ้ำ ๆ กันหลายครั้ง แต่เครื่องจะทำการสแกนแท็กที่ตั้งแสดงในผังเวลา
ด้านล่างนี้ ในผังนี้ เครื่องจะอ่านค่าการสแกนแท็กแต่ละค่าและเปรียบเทียบเงื่อนไข หากตรงตามเงื่อนไขเครื่องจะประมวลผลไม่ว่าค่าก่อนหน้านั้นจะเป็นเช่นไร

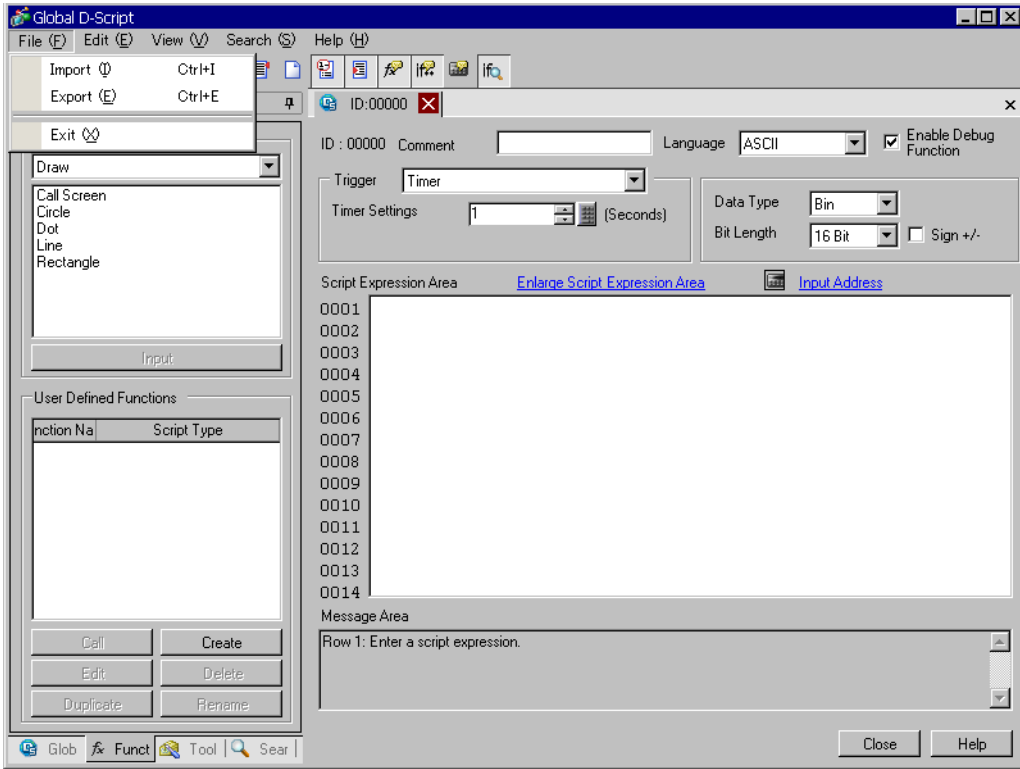
◆ ผังเวลาการประมวลผล D-Script



20.8 คำแนะนำในการตั้งค่า

20.8.1 คำแนะนำในการตั้งค่า D-Script/[Global D-Script] ทั่วไป

รายการด้านล่างนี้ คือ กล้องโต้ตอบของ [Global D-Script] ทั่วไป การตั้งค่าที่คุณสามารถระบุให้กับ D-Script จะเหมือนกับในกล้องโต้ตอบนี้ แต่จะระบุการตั้งค่า ID และทริกเกอร์ให้กับการตั้งค่า [Extended Script] ทั่วไปไม่ได้ อย่างไรก็ตาม การตั้งค่าอื่นๆ จะเหมือนกัน


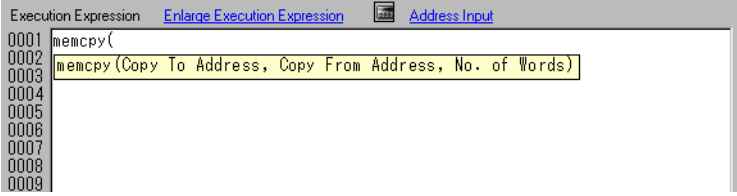



การตั้งค่า	คำอธิบาย
Export	คุณสามารถเลือกค่านี้ได้จากเมนู File Export จะเขียนสคริปต์ที่สร้างขึ้นเป็นไฟล์ข้อความ (.txt) ซึ่งสคริปต์อื่นๆ สามารถนำเข้าไปใช้ได้
Import	คุณสามารถเลือกค่านี้ได้จากเมนู File Import จะอ่านข้อมูลในสคริปต์ที่ถูกนำออก (ไฟล์ข้อความ)
Row Number	แสดงหมายเลขแถวทางด้านขวาของโปรแกรม
Auto Indent Control	ย่อหน้าข้อความคำสั่งโดยอัตโนมัติตั้งเช่นในรูปภาพด้านล่างนี้


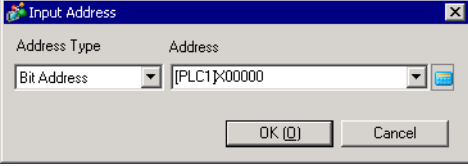
```

Script Expression Area     Enlarge Script Expression Area      Input Address
0001  if ([b:[PLC1]D00000.0]==1)
0002  {
0003      if ([b:[PLC1]D00001.0])
0004      {
0005          [b:[PLC1]D00002.0]
0006      }
0007      endif
0008  }
0009  endif
    
```


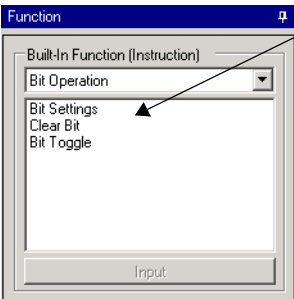
ต่อ

การตั้งค่า	คำอธิบาย
Function Input Assistance 	<p>เมื่อป้อนฟังก์ชันและวงเล็บเปิด “(” เช่นในรูปภาพด้านล่างนี้ ระบบจะแสดงรูปแบบที่มีอยู่ของฟังก์ชัน</p> 
Auto Syntax Completion 	<p>เมื่อป้อน “if” หรือ “loop” จากแป้นคีย์ ระบบจะเติมคำสั่งที่เหลือให้ครบถ้วนโดยอัตโนมัติ</p>

ต่อ

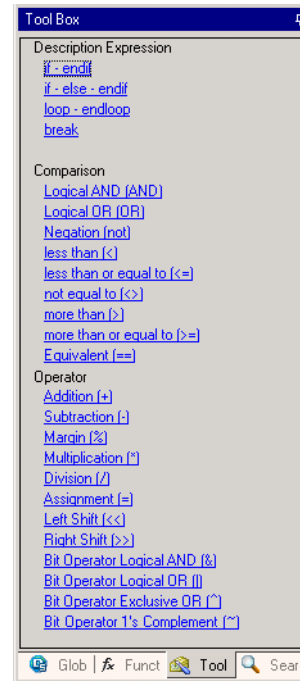
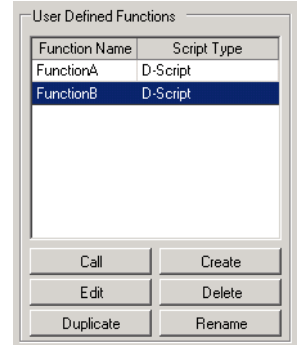
การตั้งค่า	คำอธิบาย																				
<p>Address Input </p>	<p>เมื่อสร้างสคริปต์ ให้เปิดวงเล็บเหลี่ยม ([]) เพื่อแสดงกล่องโต้ตอบ [Input Address]</p>  <p>เลือกชนิดตำแหน่งระหว่าง [Bit Address], [Word Address], [Temporary Address]</p> <ul style="list-style-type: none"> • Bit Address คุณสามารถระบุตำแหน่งอุปกรณ์/PLC, อุปกรณ์ภายในของ GP และตัวแปรบิต • Word Address คุณสามารถระบุตำแหน่งอุปกรณ์/PLC, อุปกรณ์ภายในของ GP และตัวแปรจำนวนเต็ม • Temporary Address ตำแหน่งนี้ใช้กับสคริปต์เท่านั้น <p>โปรดอ่านรายละเอียดเกี่ยวกับอุปกรณ์ภายในที่ส่วนด้านล่างนี้</p> <ul style="list-style-type: none"> ☞ “A.1.2 การสื่อสารกับอุปกรณ์/PLC โดยใช้วิธีการเชื่อมต่อโดยตรง” (หน้า A-4) ☞ “A.1.3 การใช้วิธีการเชื่อมต่อผ่านหน่วยความจำกับอุปกรณ์/PLC ที่ไม่รองรับ” (หน้า A-6) <p>ข้อสำคัญ</p> <ul style="list-style-type: none"> • ในสคริปต์ต่างๆ ห้ามตั้งค่ารหัสผ่านหรือค่าอื่นๆ โดยขึ้นต้นด้วย “0” เพราะระบบจะประมวลผลค่าตัวเลขที่ขึ้นต้นด้วย “0” เป็นข้อมูลชนิด Oct (ฐานแปด) • วิธีจำแนกรูปแบบการป้อนข้อมูลแบบต่างๆ ตัวอย่าง <table border="0" style="margin-left: 20px;"> <tr> <td>DEC (ฐานสิบ)</td> <td>: ค่าเริ่มต้นที่ไม่ใช่ศูนย์</td> </tr> <tr> <td></td> <td>ตัวอย่างเช่น 100</td> </tr> <tr> <td>Hex (ฐานสิบหก)</td> <td>: ค่าที่เริ่มต้นด้วย 0x</td> </tr> <tr> <td></td> <td>ตัวอย่างเช่น 0x100</td> </tr> <tr> <td>Oct (ฐานแปด)</td> <td>: ค่าที่เริ่มต้นด้วย 0x</td> </tr> <tr> <td></td> <td>ตัวอย่างเช่น 0100</td> </tr> </table> • ตัวอย่างการทำงานด้วยรูปแบบข้อมูลที่แตกต่างกัน โดยใช้ตัวดำเนินการ AND (Hex และ BCD) <table border="0" style="margin-left: 20px;"> <tr> <td colspan="2">Hex เท่านั้น</td> </tr> <tr> <td>0x270F & 0xFF00</td> <td>ผลลัพธ์: 0x2700</td> </tr> <tr> <td colspan="2">BCD และ Hex</td> </tr> <tr> <td>9999 & 0xFF00</td> <td>ผลลัพธ์: 0x9900</td> </tr> </table> 	DEC (ฐานสิบ)	: ค่าเริ่มต้นที่ไม่ใช่ศูนย์		ตัวอย่างเช่น 100	Hex (ฐานสิบหก)	: ค่าที่เริ่มต้นด้วย 0x		ตัวอย่างเช่น 0x100	Oct (ฐานแปด)	: ค่าที่เริ่มต้นด้วย 0x		ตัวอย่างเช่น 0100	Hex เท่านั้น		0x270F & 0xFF00	ผลลัพธ์: 0x2700	BCD และ Hex		9999 & 0xFF00	ผลลัพธ์: 0x9900
DEC (ฐานสิบ)	: ค่าเริ่มต้นที่ไม่ใช่ศูนย์																				
	ตัวอย่างเช่น 100																				
Hex (ฐานสิบหก)	: ค่าที่เริ่มต้นด้วย 0x																				
	ตัวอย่างเช่น 0x100																				
Oct (ฐานแปด)	: ค่าที่เริ่มต้นด้วย 0x																				
	ตัวอย่างเช่น 0100																				
Hex เท่านั้น																					
0x270F & 0xFF00	ผลลัพธ์: 0x2700																				
BCD และ Hex																					
9999 & 0xFF00	ผลลัพธ์: 0x9900																				

ต่อ

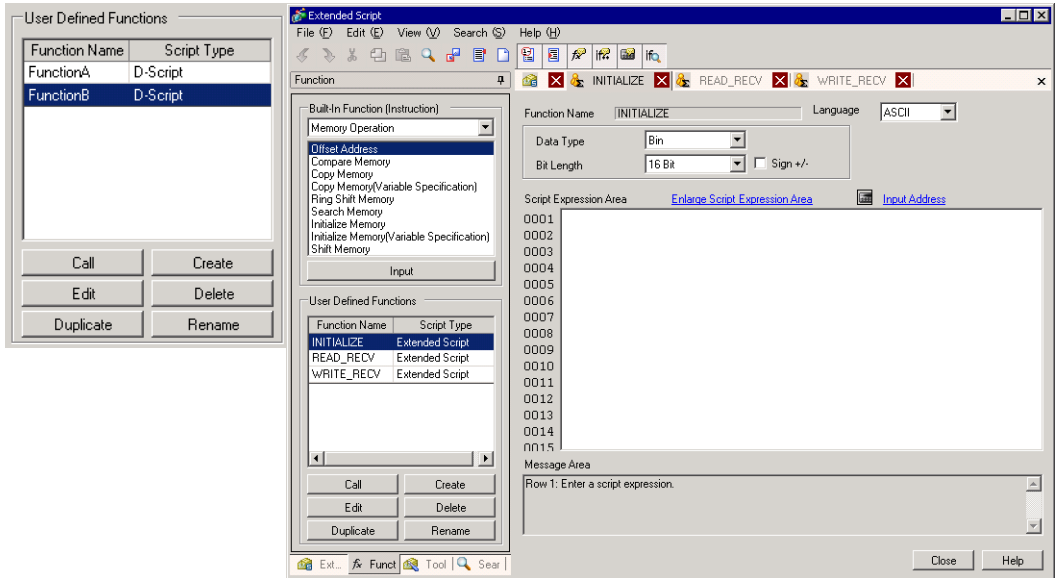
การตั้งค่า	คำอธิบาย
Auto Syntax Analysis 	ตรวจสอบไวยากรณ์ระหว่างสร้างสคริปต์ ผลการตรวจสอบจะแสดงอยู่ที่ด้านล่างของหน้าต่าง <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> Message Area Row 5: A statement is required in { } of an 'If' statement. Row 5: The expression is incorrect. </div>
ID	สคริปต์จะถูกจัดการตามหมายเลข ID เมื่อสร้างสคริปต์หลายสคริปต์โดยมีเงื่อนไขการทริกเกอร์ที่ต่างกัน ให้ตั้งค่าตั้งแต่ 0 ถึง 65,535
Comment	ใส่คำอธิบายสคริปต์
Language	เลือกภาษาจากรายการดรอปดาวน์ ซึ่งได้แก่ [ASCII], [Japanese], [Chinese (Traditional)], [Chinese (Simplified)] หรือ [Korean]
Enable Debug Function	กำหนดว่าจะใช้งานฟังก์ชันดีบักหรือไม่ หากมีฟังก์ชัน <code>_debug</code> อยู่ในส่วนเนื้อหาของสคริปต์ ฟังก์ชัน <code>_debug</code> จะทำงาน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฟังก์ชันนี้ โปรดดูที่ “21.7.1 ฟังก์ชัน Debug” (หน้า 21-65)
เงื่อนไขการทริกเกอร์	ตั้งค่าเงื่อนไขการทริกเกอร์ที่จะทำให้สคริปต์ทำงาน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฟังก์ชันนี้ โปรดดูที่ “20.7 การตั้งค่าเงื่อนไขการทริกเกอร์” (หน้า 20-46) Extended Script จะไม่มีการตั้งค่าเงื่อนไขการทริกเกอร์
Data Type	ตั้งค่ารูปแบบข้อมูลสำหรับสคริปต์เป็น Bin หรือ BCD สำหรับ Extended Script จะกำหนดค่าไว้ตายตัวคือ Bin
Bit Length	ตั้งค่าความยาวข้อมูลสำหรับสคริปต์ระหว่าง 16 Bit หรือ 32 Bit
Sign +/-	เลือกช่องนี้เมื่อคุณต้องการแทรกจำนวนลบ คุณสามารถเลือกช่องนี้ได้เฉพาะเมื่อตั้งค่า Data Type เป็น Bin เท่านั้น
Execution Expression	รายละเอียดของสคริปต์
Built-in Function (Instruction)	เลือกคำสั่งและฟังก์ชันที่จะเพิ่มในสคริปต์ได้ง่ายยิ่งขึ้นจากแถบเครื่องมือ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งและฟังก์ชันต่างๆ ที่ใช้ได้ โปรดดูที่ “21.13 รายการคำสั่ง” (หน้า 21-100) Built-in Function <div style="display: flex; align-items: flex-start; margin-top: 10px;"> <div style="flex: 1;">  </div> <div style="flex: 2; padding-left: 10px;"> <p>เลือกประเภทจาก [Built-In Function (Instruction)] ฟังก์ชันที่เกี่ยวข้อง จะแสดงอยู่ในพื้นที่ส่วนล่างสุด เลือกฟังก์ชันและคลิกที่ [Input] กล้องได้ตอบการตั้งค่าที่เกี่ยวข้องจะปรากฏขึ้น</p> </div> </div>

ต่อ

การตั้งค่า	คำอธิบาย
<p>User Defined Function</p>	<p>ลงทะเบียนสคริปต์เป็นฟังก์ชันที่กำหนดโดยผู้ใช้ ซึ่งสคริปต์อื่นๆ สามารถใช้งานได้</p> <p>หมายเหตุ</p> <ul style="list-style-type: none"> โปรดดูรายละเอียดเพิ่มเติมเกี่ยวกับฟังก์ชันที่กำหนด โดยผู้ใช้ที่ “20.8.2 คำแนะนำในการตั้งค่า User-Defined Functions” (หน้า 20-58)
<p>Tool Box</p>	<p>ใช้เป็นทางลัด โดยใช้สำหรับเลือกคำสั่งที่จะใช้ในสคริปต์ จาก Toolbox และยังสามารถเลือกคำสั่งเช่น ค้นหาและจัดตำแหน่ง ข้อความที่ใช้ในสคริปต์ได้อีกด้วย</p> <p>โปรดดูข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งที่มีอยู่ได้ที่ “บทที่ 21 คำสั่งและคำอธิบาย” (หน้า 21-1)</p>



20.8.2 คำแนะนำในการตั้งค่า User-Defined Functions



การตั้งค่า	คำอธิบาย
Call	เรียกฟังก์ชันที่สร้างขึ้น โดยเลือกฟังก์ชันที่จะเรียกใช้งาน คลิก [Call] และ “ชื่อฟังก์ชันที่เรียกใช้” จะถูกใส่ลงในฟิลด์ Execution Expression
Create	สร้างฟังก์ชันใหม่ คลิกที่ [Create] กล่องโต้ตอบ [Function Name] จะปรากฏขึ้น
Edit	แก้ไขฟังก์ชันที่มีอยู่ โดยเลือกฟังก์ชันที่จะแก้ไข แล้วคลิก [Edit] กล่องโต้ตอบ [D-Script Function] จะปรากฏขึ้น
Delete	ลบฟังก์ชันที่มีอยู่ โดยเลือกฟังก์ชันที่จะลบ แล้วคลิก [Delete]
Duplicate	คัดลอกฟังก์ชันที่สร้างขึ้น โดยเลือกฟังก์ชันที่จะคัดลอกแล้วคลิก [Duplicate] และเมื่อกล่องโต้ตอบ [Input Function Name] ปรากฏขึ้น ให้ป้อนชื่อฟังก์ชันชื่อใหม่
Rename	เปลี่ยนชื่อของฟังก์ชันที่มีอยู่ คลิก [Rename] กล่องโต้ตอบ Rename Function จะปรากฏขึ้น

20.9 ข้อจำกัด

20.9.1 ข้อจำกัดของ D-Script/Global D-Script

- ในการเขียนโปรแกรม D-Script โปรดจำไว้ว่าตำแหน่งสามตำแหน่งจะใช้หน่วยความจำเท่ากับพาร์ทหนึ่งพาร์ท จำนวนตำแหน่งสูงสุดที่สามารถใช้กับ D-Script ได้คือ 255 ตำแหน่ง แต่ควรใช้ตำแหน่งให้น้อยที่สุดเท่าที่ทำได้ เนื่องจากยิ่งใช้อุปกรณ์มากเท่าไร ก็จะทำให้การตอบสนองช้าลงเท่านั้น
- คุณสมบัติ Convert Address ไม่สามารถแปลงตำแหน่งที่ใช้ใน D-Script ได้ ให้เปิด D-Script Editor เพื่อแก้ไขตำแหน่งเหล่านี้
- เมื่อเปลี่ยนแปลงการตั้งค่า Connected Device Type จากเมนู Project Manager > Project > Save As จะไม่สามารถแก้ไขตำแหน่งที่ D-Script ใช้ได้ ให้ใช้กล่องโต้ตอบ D-Script Settings ในการเปลี่ยนแปลงตำแหน่ง
- D-Script ไม่สามารถคำนวณค่าฟลอยต์ตั้งพอยต์ (Float Variable หรือ Real Variable) หรือตัวแปรโครงสร้างได้ อย่างไรก็ตาม คุณสามารถทำการคำนวณอิลิเมนต์แต่ละค่าได้จากตัวแปรโครงสร้าง
- ขนาดของ D-Script จะมีผลต่อเวลาสำหรับการแสดงผล โปรดทราบว่าการใช้ตำแหน่งเป็นจำนวนมากทำให้ประสิทธิภาพการทำงานของโปรแกรมลดลงอย่างเห็นได้ชัด
- ห้ามระบุ [Continuous Action] ในเงื่อนไขการทริกเกอร์ให้กับสคริปต์เพื่อเขียนตำแหน่งอุปกรณ์/PLC เพราะจะเกิดข้อผิดพลาดเนื่องจากการประมวลผลการสื่อสารไม่สามารถจัดการกับคำสั่งเขียนข้อมูลจำนวนมากได้ หากต้องการใช้ [Continuous Action] ให้ใช้อุปกรณ์ภายในหรือตำแหน่งชั่วคราวของ GP
- เมื่อเรียกฟังก์ชันหนึ่งจากภายในฟังก์ชันหนึ่ง สามารถเรียกได้สูงสุด 9 ระดับ (ซ้อนกัน) ห้ามสร้างมากกว่า 9 ระดับ
- คุณสามารถเรียกฟังก์ชันซ้อนกันได้ไม่เกิน 9 ระดับ
- คุณสามารถสร้างฟังก์ชันได้ไม่เกิน 254 ฟังก์ชัน

- ◆ การทำงานของ D-script ที่เรียกใช้โดยทริกเกอร์หลังจากหน้าจอเปลี่ยนแปลงไป จะเป็นดังต่อไปนี้ โดยจะเป็นไปตามอุปกรณ์ที่ระบุให้กับเงื่อนไขทริกเกอร์

เงื่อนไขการทริกเกอร์	อุปกรณ์เชื่อมต่อหรืออุปกรณ์ภายในของ GP (LS/USR)				การเชื่อมต่อผ่านหน่วยความจำ			
	ค่าปัจจุบันหรือเงื่อนไข	Bit "0"	Bit "1"	ไม่ตรงตามเงื่อนไข	ตรงตามเงื่อนไข	Bit "0"	Bit "1"	ไม่ตรงตามเงื่อนไข
ขอบขาขึ้นของบิต	Disable	Enable	—	—	Disable	Disable	—	—
ขอบขาลงของบิต	Enable	Disable	—	—	Disable	Disable	—	—
Bit Change	Enable	Enable	—	—	Disable	Disable	—	—
การตั้งค่าตัวตั้งเวลา	Disable	Disable	Disable	Disable	Disable	Disable	Disable	Disable
ตรวจพบว่าเป็นจริง	—	—	Disable	Enable	—	—	Disable	Enable
ตรวจพบว่าเป็นเท็จ	—	—	Enable	Disable	—	—	Enable	Disable

ทำงานทันทีหลังจากเปลี่ยนแปลงหน้าจอหรือเปิดเครื่อง

ไม่ทำงานทันทีหลังเปลี่ยนแปลงหน้าจอหรือเมื่อเปิดเครื่อง

- เมื่อตัวตั้งเวลาทำงาน ตัวตั้งเวลาจะเริ่มนับเวลาที่หลังจากหน้าจอเปลี่ยนแปลง
- เมื่อใช้ Global D-Script จะมีการทำงานต่าง ๆ ดังที่กล่าวถึงด้านบนนี้เฉพาะเมื่อเปิดเครื่อง GP เท่านั้น อย่างไรก็ตาม เมื่อหน้าจอ GP เปลี่ยนแปลงจะไม่มีการทำงานต่าง ๆ ดังที่กล่าวถึงด้านบน และจอมอนิเตอร์จะทำงานโดยใช้เงื่อนไขการทริกเกอร์ที่ตั้งค่าไว้
- เมื่อมีตัวตั้งเวลาอยู่ใน Global D-Script ตัวตั้งเวลาจะเริ่มนับเวลาที่หลังจากเปิดเครื่อง GP

หมายเหตุ

- ห้ามใช้ปุ่มบนหน้าจอสัมผัสตั้งค่าทริกเกอร์บิตหรือสั่งงานบิตเริ่มต้นในโปรแกรม ระยะเวลาในการแตะเพื่อป้อนข้อมูลอาจไม่ถูกต้อง ทำให้บิตที่ป้อนไม่ถูกต้อง

- ◆ เมื่อกำหนดค่าให้ตำแหน่งสำหรับเปลี่ยนหน้าจอในขณะที่กำลังเรียกใช้คำสั่ง D-Script ระบบจะเปลี่ยนหน้าจอหลังจากประมวลผล D-Scripts ทั้งหมดแล้ว

ตัวอย่าง

ID	00000				
Data Type	Bin	Data Length	16 Bit	Sign +/-	None
Trigger	Leading Bit([b:M0000])				
	[w:[PLC1]D0100]=0		// (1)		
	[w:[#INTERNAL]LS0008]=30		// (2)	Switches to Base screen Number 30	
	[w:[PLC1]D0101]=1		// (3)		
	[w:[PLC1]D0102]=2		// (4)		

เมื่อ D-Script ข้างต้นถูกเรียกใช้ ระบบจะเปลี่ยนหน้าจอหลังจากประมวลผล (3) และ (4) แล้ว

- ◆ เมื่อตั้งค่าข้อมูลที่ให้กับ D-Script ด้วยการแตะจากเครื่อง GP ให้ตรวจสอบว่าได้เขียนข้อมูลทั้งหมดแล้ว แล้วจึงเรียกใช้ D-Script

◆ ข้อจำกัดเฉพาะของ Global D-Script

- เมื่อเปิดเครื่อง GP จะไม่มีการทำงานตามที่แสดงในตารางในหน้าที่แล้ว เมื่อหน้าจอเปลี่ยนแปลง จะไม่นำตารางดังกล่าวมาใช้ และเงื่อนไขการทริกเกอร์จะถูกตรวจสอบอย่างต่อเนื่อง
- Global D-Script จะถูกพักการทำงานชั่วคราวในระหว่างเปลี่ยนหน้าจอหรือมีการทำงานอื่น ๆ ของเครื่อง GP
- หลังจากเปิดเครื่อง GP Global D-Script จะยังไม่ทำงาน จนกว่าเครื่องจะอ่านข้อมูลทั้งหมดของหน้าจอเริ่มต้นเสร็จแล้ว อย่างไรก็ตาม หลังจากหน้าจอเริ่มต้นเปลี่ยนไป Global D-Script อาจทำงานก่อนที่จะอ่านข้อมูลเสร็จ
- จำนวนอุปกรณ์สูงสุดใน Global D-Script คือ 255 หากมีจำนวนอุปกรณ์เกินจากที่กำหนดไว้ D-Script จะไม่ทำงาน เนื่องจากอุปกรณ์เหล่านี้จะอ่านข้อมูลเสมอโดยไม่คำนึงถึงหน้าจอ จึงควรตั้งค่าจำนวนอุปกรณ์ใน D-Script ให้มีจำนวนน้อยที่สุด มิฉะนั้น ประสิทธิภาพในการทำงานจะลดลง
- จำนวน Global D-Scripts สูงสุดที่สามารถใช้ได้คือ 32 ฟังก์ชันที่ใช้อยู่ในขณะนี้ก็นับเป็นหนึ่งใน Global D-Script เช่นกัน เมื่อ Global D-Scripts มีจำนวนครบ 32 ระบบจะไม่สนใจ Global D-Scripts อื่นได้อีก

◆ ข้อจำกัดในการทำงานของพอร์ต SIO

- ตำแหน่งที่กำหนดในฟังก์ชัน Send/Receive จะไม่รวมอยู่ในจำนวนตำแหน่ง D-Script
- Control เป็นตัวแปรแบบเขียนข้อมูลอย่างเดียว ส่วน Status และ Received Data เป็นตัวแปรแบบอ่านอย่างเดียว การอ่านข้อมูลตัวแปร Control หรือการเขียนข้อมูลลงในตัวแปร Status จะทำให้การทำงานล้มเหลว
- ให้สร้าง D-Scripts (หรือฟังก์ชัน) ต่างหากสำหรับการส่งและการรับ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฟังก์ชันถ่ายโอนข้อมูล โปรดดูที่
☞ “ ■ ฟังก์ชันการทำงาน ” (หน้า 20-24)
- พื้นที่สำหรับผู้ใช้ในอุปกรณ์ LS (LS20 ถึง LS2031 และ LS2096 ถึง LS8191) สามารถจัดเก็บข้อมูลของฟังก์ชัน Send/Receive ได้
- ใน [System Settings] - [Script Settings] หากไม่ได้ตั้งค่า [D-Script/Global D-Script] ไว้ บิตที่ 13 ของ LS2032 จะเปลี่ยนเป็นสถานะเปิด เมื่อดำเนินการ Readout กับฟังก์ชัน Send, ฟังก์ชัน Receive, ตัวแปร Control, ตัวแปร Status และคุณสมบัติ Received Data Size สำหรับข้อมูลเกี่ยวกับโครงสร้างของอุปกรณ์ภายใน โปรดดูที่
☞ “A.1.4.3 รีเลย์พิเศษ” (หน้า A-19)
- เมื่อใช้ฟังก์ชัน Send/Receive ให้ตั้งค่าความยาวบิตของ D-Script เป็น 16 บิต โปรดทราบว่าการทำงานจะล้มเหลวหากตั้งค่าความยาวบิตเป็น 32 บิต
- บัฟเฟอร์การส่งข้อมูลมีขนาดเท่ากับ 2048 ไบต์ ส่วนบัฟเฟอร์การรับข้อมูลมีขนาด 8192 ไบต์ สัญญาณ ER (เอาต์พุต) สัญญาณ RS (เอาต์พุต) จะถูกปิด หากบัฟเฟอร์การรับข้อมูลรับข้อมูลเข้ามาแล้วอย่างน้อย 80%

◆ ข้อจำกัดของการทำงานด้วยรูปแบบข้อมูล BCD

หากโปรแกรมพบค่าที่ไม่สามารถแปลงเป็นรูปแบบ BCD ในระหว่างการทำงาน โปรแกรมจะหยุดทำงานค่าเหล่านี้ได้แก่ A ถึง F ของข้อมูลแบบเลขฐานสิบหก ห้ามใช้ค่าเหล่านี้

หากโปรแกรมหยุดทำงานเนื่องจากค่าที่ไม่ใช่รูปแบบ BCD บิต 7 ในข้อมูลรีเลย์ร่วม (LS2032) ในเครื่อง GP จะเปิดขึ้น บิตนี้จะไม่ปิดจนกว่าจะปิดเครื่อง GP หรือเครื่อง GP เข้าสู่โหมดออฟไลน์

ตัวอย่าง

$$[w:[PLC1]D0200]=([w:[PLC1]D0300]<<2)+80$$

หาก D300 เท่ากับ 3 การเลื่อนบิตสองบิตไปทางซ้ายจะทำให้เกิดผลลัพธ์เป็น 0x000C ซึ่งไม่สามารถแปลงเป็นรูปแบบ BCD ได้ และจะขัดจังหวะการดำเนินการของโปรแกรม

$$[w:[PLC1]D0200]=[w:[PLC1]D0300]<<2$$

หาก D300 เท่ากับ 3 การเลื่อนบิตสองบิตไปทางซ้ายจะทำให้เกิดผลลัพธ์เป็น 0x000C แต่ต่างจากตัวอย่างด้านบนตรงที่ 0x000C เป็นผลลัพธ์จากการทำงานที่จะจัดเก็บไว้ในหน่วยความจำ และไม่ทำให้โปรแกรมหยุดทำงาน

◆ ข้อจำกัดของการทำงานด้วยค่าศูนย์

หากคุณใช้ศูนย์เป็นตัวหารใน division (/) และ remainder (%) โปรแกรมจะหยุดดำเนินการ ห้ามใช้ศูนย์เป็นตัวหาร

หากโปรแกรมหยุดทำงานเนื่องจากค่าที่ไม่ใช่ BCD บิต 8 ในข้อมูลรีเลย์ร่วม (LS2032) ในเครื่อง GP จะเปิดขึ้น บิตนี้จะไม่ปิดจนกว่าจะปิดเครื่อง GP หรือเครื่อง GP เข้าสู่โหมดออฟไลน์

◆ ข้อควรทราบเกี่ยวกับความล่าช้าในระหว่างการทำงานของคำสั่ง Assign

การใช้ตำแหน่งอุปกรณ์ในคำสั่ง Assign อาจทำให้การเขียนข้อมูลล่าช้าได้ เนื่องจาก GP ต้องอ่านข้อมูลตำแหน่งจากอุปกรณ์ที่เชื่อมต่ออยู่ โปรดดูตัวอย่างด้านล่างนี้

$$[w:[PLC1]D0200]=[w:[PLC1]D0300]+1 \dots$$

$$[w:[PLC1]D0201]=[w:[PLC1]D0200]+1 \dots$$

ข้อความคำสั่ง (1) กำหนด (D0300+1) ลงใน D0200 แต่ในข้อความคำสั่ง (2) ผลลัพธ์ของข้อความคำสั่ง (1) ไม่ได้ถูกกำหนดลงใน D0200 เนื่องจากการสื่อสารกับอุปกรณ์/PLC ใช้เวลานาน ในกรณีเช่นนี้ ให้เขียนโปรแกรมให้จัดเก็บผลลัพธ์ของข้อความคำสั่ง (1) ในพื้นที่ LS ก่อนที่จะดำเนินการกับผลลัพธ์นั้น ดังเช่นตัวอย่างด้านล่างนี้

$$[w:[\#INTERNAL]LS0100]=[w:[PLC1]D0300]+1$$

$$[w:[PLC1]D0200]=[w:[\#INTERNAL]LS0100]$$

$$[w:[PLC1]D0201]=[w:[\#INTERNAL]LS0100]+1$$

20.9.2 ข้อจำกัดของ Extended Script

- ตำแหน่งอุปกรณ์ต่าง ๆ สามารถใช้ได้เฉพาะพื้นที่ LS และพื้นที่ USR (พื้นที่เสริมสำหรับผู้ใช้) เท่านั้น
- ตำแหน่งชั่วคราวของ D-Scripts และ Global D-Scripts จะถูกแยกจัดการจากตำแหน่งชั่วคราวของ Extended Script ดังนั้น การเปลี่ยนแปลงต่าง ๆ ในตำแหน่งชั่วคราวของ D-Scripts และ Global D-Scripts จึงไม่ปรากฏในตำแหน่งชั่วคราวของ Extended Script
- คุณสามารถเรียกฟังก์ชันที่กำหนดโดยผู้ใช้ที่สร้างด้วย D-Script/Global D-Script ได้ แต่หากคุณเข้าใช้ตำแหน่งอุปกรณ์ที่อยู่นอกอุปกรณ์ภายในที่มีอยู่ในฟังก์ชัน ฟังก์ชันอาจทำงานไม่ถูกต้อง นอกจากนี้ เมื่อมีการถ่ายโอนข้อมูล (ในระหว่างสร้างข้อมูลสำหรับ GP) ระบบจะสร้างฟังก์ชันที่กำหนดโดยผู้ใช้ของ D-Scripts, Global D-Scripts และ Extended Script แยกต่างหากจากกัน
- เมื่อเรียกฟังก์ชันหนึ่งจากฟังก์ชันหนึ่ง สามารถเรียกได้สูงสุด 9 ระดับ (การซ้อนกัน)
- คุณสามารถเรียกฟังก์ชันได้ไม่เกิน 254 ฟังก์ชัน (จำนวนฟังก์ชันที่ใช้กับฟังก์ชัน "Call" ได้คือ 254)
- Extended Script จะไม่มีผลต่อการนับจำนวนแท็ก
- ฟังก์ชันที่รองรับเฉพาะ Extended Script เช่น การทำงานของสตริง จะไม่ทำงานหากเรียกด้วย D-Script หรือ Global D-Script
- รูปแบบข้อมูลที่สามารถใช้ได้คือ Bin ข้อมูลที่เป็นรูปแบบ BCD ไม่สามารถใช้ได้
- บัฟเฟอร์การส่งข้อมูลมีขนาดเท่ากับ 2048 ไบต์ ส่วนบัฟเฟอร์การรับข้อมูลมีขนาด 8192 ไบต์ บรรทัด CTS จะถูกปิด หากบัฟเฟอร์การรับข้อมูลรับข้อมูลเข้ามาแล้วอย่างน้อย 80%

- ไม่สามารถเลือกโปรโตคอลทั่วไปและ Extended Script พร้อมกันได้ ตารางต่อไปนี้แสดงข้อมูลเพิ่มเติมเกี่ยวกับการใช้ร่วมกัน

การตั้งค่า Extended SIO	D-Script/ Global D-Script ฟังก์ชัน Extended SIO ของ Extended Script	ฟังก์ชัน Extended SIO ของ Extended Script
โปรโตคอลทั่วไป	รองรับ: ทำงานได้	ไม่รองรับ: ทำงานไม่ได้
Extended Script	ไม่รองรับ: ทำงานไม่ได้	รองรับ: ทำงานได้

- หลักเกณฑ์ในการตั้งค่าสตริงอักขระ

เมื่อใช้สตริงอักขระที่มี "_strset ()" และฟังก์ชันอื่นๆ ให้ใส่สตริงอักขระไว้ในเครื่องหมายอัปประกาศ ("") เมื่อต้องการแสดงเครื่องหมายอัปประกาศในสตริงอักขระ ให้ใส่สัญลักษณ์ "\" และแสดงเป็น [N] ไม่มีวิธีใดที่จะเขียนแทนสัญลักษณ์ "\" ตัวเดียวได้ หากจำเป็น ให้ใช้การตั้งค่ารูปแบบรหัสอักขระ (_strset (databuf0, 92))

ตัวอย่าง

```
"ABC"DEF" → ABC"DEF
"ABCDEF" → ABCDEF
"ABC\"DEF" → ABC\"DEF
"ABC\DEF" → ABC\DEF
```

- ◆ ตารางต่อไปนี้จะแสดงขนาดของบัฟเฟอร์สำหรับ Extended SIO, databuf0, databuf1, databuf2 และ databuf3

บัฟเฟอร์	ชื่อบัฟเฟอร์	ขนาดอักขระ
บัฟเฟอร์ข้อมูล 0	databuf0	1KB
บัฟเฟอร์ข้อมูล 1	databuf1	1KB
บัฟเฟอร์ข้อมูล 2	databuf2	1KB
บัฟเฟอร์ข้อมูล 3	databuf3	1KB

20.9.3 ข้อจำกัดของฟังก์ชันที่กำหนดโดยผู้ใช้

- ส่วนของคำสั่งที่สามารถใช้ได้จะแตกต่างกันไปในแต่ละสคริปต์ เมื่อใช้คำสั่ง โปรดดูที่ “21.13 รายการคำสั่ง” (หน้า 21-100)
- สำหรับชื่อฟังก์ชัน คุณสามารถใช้ตัวอักษรภาษาอังกฤษตัวใดก็ได้หรือใช้เส้นใต้อักขระ "_" (แต่ชื่อฟังก์ชันต้องขึ้นต้นด้วยอักขระตัวเลขผสมตัวอักษร)
- ห้ามใช้ชื่อต่อไปนี้เป็นชื่อฟังก์ชัน

and	b_call	Bcall	_bin2hexasc	break	Call
_CF_delete	_CF_dir	_CF_read	_CF_read_csv	_CF_rename	_CF_write
clear	databuf0	databuf1	databuf2	databuf3	_decasc2bin
_dlcopy	dsp_arc	dsp_circle	dsp_dot	dsp_line	dsp_rectangle
else	endif	fall	_hexasc2bin	if	IO_READ
IO_READ_EX	IO_READ_WAIT	IO_WRITE	IO_WRITE_EX	loop	_memcmp
memcpy	_memcpy_EX	memring	_memsearch	memset	_memset_EX
_memshift	not	or	return	rise	rise_expr
set	_strcat	_strlen	_strmid	_strset	timer
toggle	_wait				

20.9.4 ข้อควรทราบเกี่ยวกับผลการทำงาน

■ Overflowing Digit

Overflowing Digit เกิดขึ้นจากการคำนวณที่ถูกปิดเศษ

เมื่อทำการคำนวณข้อมูล 16 บิตที่ไม่ได้ระบุเครื่องหมาย:

- $65535 + 1 = 0$ (เกิด Overflowing Digit)
- $(65534 * 2) / 2 = 32766$ (เกิด Overflowing Digit)
- $(65534 / 2) * 2 = 65534$ (ไม่เกิด Overflowing Digit)

■ ความแตกต่างของการคำนวณเศษเหลือ

ผลลัพธ์ของการคำนวณเศษเหลือจะขึ้นอยู่กับว่ามีการใส่เครื่องหมายทางฝั่งซ้ายและขวาหรือไม่

- $-9 \% 5 = -4$
- $9 \% -5 = 4$

■ การปัดจุดทศนิยม

จุดทศนิยมที่เป็นผลจากการหารจะถูกปิดเศษ

- $10 / 3 * 3 = 9$
- $10 * 3 / 3 = 10$

■ ข้อควรทราบในการทำงานด้วยข้อมูล BCD

การทำงานกับข้อมูล BCD ที่ทำให้เกิด Overflowing Digit จะมีผลลัพธ์ที่ไม่ถูกต้อง

20.9.5 ข้อผิดพลาด

ข้อความแสดงข้อผิดพลาดต่อไปนี้จะปรากฏขึ้นเมื่อตั้งค่าสคริปต์ไม่ถูกต้อง ข้อผิดพลาดจะแสดงอยู่ที่ด้านล่างของหน้าจอเครื่อง GP

รหัสข้อผิดพลาดจะถูกเขียนลงในตำแหน่ง LS91XX ตัวเลขที่เขียนในรหัสข้อผิดพลาดจะเป็นตัวเลขที่ตามหลัง RAAA ในตารางด้านล่างนี้ (ตัวอย่างเช่น เมื่อเกิดข้อผิดพลาด RAAA130 ขึ้น ระบบจะเขียนเลข '130' ไว้)

รายการแสดงรหัสข้อผิดพลาดของสคริปต์

D-Script (ตำแหน่งข้อผิดพลาด = LS9120)	Global D-Script (ตำแหน่งข้อผิดพลาด = LS9110)	Extended Script (ตำแหน่งข้อผิดพลาด = LS9100)
-	RAAA130	RAAA140
ไม่ใช้งาน	Global D-Script Error. (The Total Number of Global D-Scripts exceeds the maximum of 32.)	Extended D-Script Error (The total no. of functions exceeds the max of 255.)
-	RAAA131	-
ไม่ใช้งาน	Global D-Script Error. (The total no. of devices exceeds the maximum of 255.)	ไม่ใช้งาน
RAAA120	RAAA132	RAAA141
D-Script Error (The specified function does not exist or the function has an error.)	Global D-Script Error (The specified function does not exist or the function has an error.)	Extended D-Script Error (The specified function does not exist or the function has an error.)
RAAA121	RAAA133	RAAA142
D-Script Error (These functions are nested to 10 levels or more.)	Global D-Script Error (These functions are nested to 10 levels or more.)	Extended D-Script Error (These functions are nested to 10 levels or more.)
RAAA122	RAAA134	RAAA143
D-Script Error (An expression exists, that is not supported by this version.)	Global D-Script Error (An expression exists, that is not supported by this version.)	Extended D-Script Error (An expression exists, that is not supported by this version.)
RAAA123	RAAA135	RAAA144
D-Script Error (The SIO operation function is used in a condition where no device/PLC has been set.)	Global D-Script Error (The SIO operation function is used in a condition where no device/PLC has been set.)	Extended D-Script Error (The SIO operation function is used in a condition where no device/PLC has been set.)
RAAA124	RAAA136	RAAA145
The D-Script has an error.	The Global D-script has an error.	The Extended D-Script has an error.

