

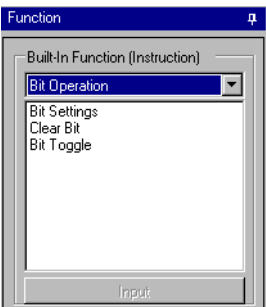
# 21

## คำสั่งและนิพจน์ ของโปรแกรม

ในบทนี้จะอธิบายเกี่ยวกับ “คำสั่งที่ใช้ในสคริปต์” ของ GP-Pro EX  
สำหรับวิธีการเขียนโปรแกรมโดยใช้สคริปต์ โปรดดูที่ “บทที่ 20 การเขียนโปรแกรมโดยใช้สคริปต์  
(การเขียนโปรแกรมที่ไม่ใช้พาร์ท)” (หน้า 20-1)

|       |                               |       |
|-------|-------------------------------|-------|
| 21.1  | การทำงานของบิต .....          | 21-2  |
| 21.2  | การวาด .....                  | 21-3  |
| 21.3  | การทำงานของหน่วยความจำ .....  | 21-7  |
| 21.4  | การทำงานของพอร์ต SIO .....    | 21-26 |
| 21.5  | การใช้งานไฟล์ในการ์ด CF ..... | 21-38 |
| 21.6  | การทำงานของเครื่องพิมพ์ ..... | 21-60 |
| 21.7  | อื่นๆ .....                   | 21-65 |
| 21.8  | นิพจน์คำอธิบาย .....          | 21-67 |
| 21.9  | การเปรียบเทียบ .....          | 21-71 |
| 21.10 | ตัวดำเนินการ .....            | 21-73 |
| 21.11 | การทำงานของข้อความ .....      | 21-77 |
| 21.12 | ตัวอย่างการทำงาน .....        | 21-94 |
| 21.13 | รายการคำสั่ง .....            | 21-98 |

## 21.1 การทำงานของบิต

| Bit Operation                                                                     | ข้อมูลสรุปของฟังก์ชัน                                                                                  |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
|  | Bit Settings<br>“21.1.1 Bit Settings” (หน้า 21-2)<br>เปลี่ยนตำแหน่งบิตที่ระบุจาก 0 เป็น 1              |
|                                                                                   | Clear Bit<br>“21.1.2 Clear Bit” (หน้า 21-2)<br>เปลี่ยนตำแหน่งบิตที่ระบุจาก 1 เป็น 0                    |
|                                                                                   | Bit Toggle<br>“21.1.3 Bit Toggle” (หน้า 21-2)<br>เปลี่ยนตำแหน่งบิตที่ระบุจาก 1 เป็น 0 หรือจาก 0 เป็น 1 |

### 21.1.1 Bit Settings

| รายการ     | คำอธิบาย                             |
|------------|--------------------------------------|
| ข้อมูลสรุป | เปลี่ยนตำแหน่งบิตที่ระบุจาก 0 เป็น 1 |
| รูปแบบ     | set ( )                              |

#### ตัวอย่างนิพจน์

set ([b:[#INTERNAL]LS010000])

จากตัวอย่างข้างบน บิตที่ 00 ของ LS0100 ถูกเปลี่ยนจาก 0 เป็น 1

### 21.1.2 Clear Bit

| รายการ     | คำอธิบาย                             |
|------------|--------------------------------------|
| ข้อมูลสรุป | เปลี่ยนตำแหน่งบิตที่ระบุจาก 1 เป็น 0 |
| รูปแบบ     | clear ( )                            |

#### ตัวอย่างนิพจน์

clear ([b:[#INTERNAL]LS010000])

จากตัวอย่างข้างบน บิตที่ 00 ของ LS0100 ถูกเปลี่ยนจาก 1 เป็น 0

### 21.1.3 Bit Toggle

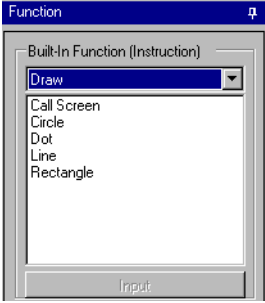
| รายการ     | คำอธิบาย                                              |
|------------|-------------------------------------------------------|
| ข้อมูลสรุป | เปลี่ยนตำแหน่งบิตที่ระบุจาก 1 เป็น 0 หรือจาก 0 เป็น 1 |
| รูปแบบ     | toggle ( )                                            |

#### ตัวอย่างนิพจน์

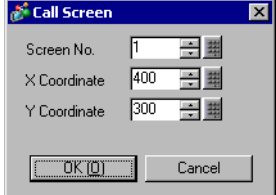
toggle ([b:[#INTERNAL]LS010000])

จากตัวอย่างข้างบน บิตที่ 00 ของ LS0100 ถูกเปลี่ยนจาก 1 เป็น 0 หรือจาก 0 เป็น 1

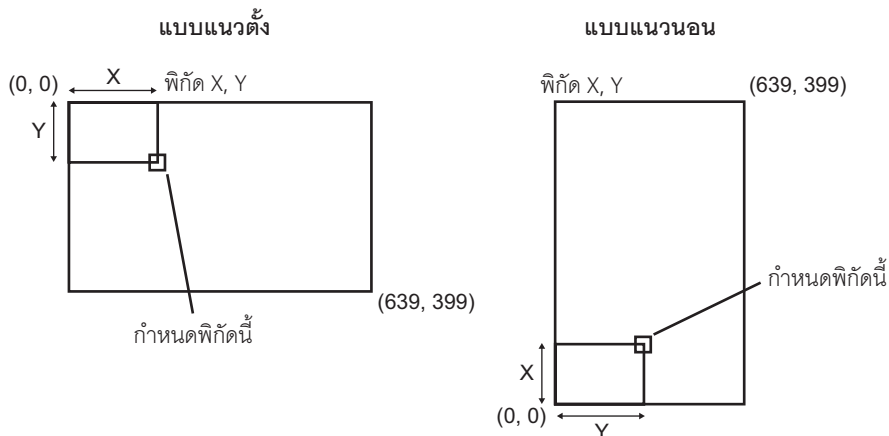
## 21.2 การวาด

| Draw                                                                              | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                        |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Call Screen</p> <p>☞ “21.2.1 Call Screen” (หน้า 21-3)<br/>เรียกหน้าจอ (หน้าจอหลัก) ที่มีหมายเลขหน้าจอที่กำหนด ฟังก์ชันนี้ใช้ใน Extended Script ไม่ได้</p> |
|                                                                                   | <p>Circle</p> <p>☞ “21.2.2 Circle” (หน้า 21-4)<br/>วาดวงกลมที่กำหนด</p>                                                                                      |
|                                                                                   | <p>Dot</p> <p>☞ “21.2.3 Dot” (หน้า 21-5)<br/>วาดจุดที่กำหนด</p>                                                                                              |
|                                                                                   | <p>Line</p> <p>☞ “21.2.4 Line” (หน้า 21-5)<br/>วาดเส้นที่กำหนด</p>                                                                                           |
|                                                                                   | <p>Rectangle</p> <p>☞ “21.2.5 Rectangle” (หน้า 21-6)<br/>วาดสี่เหลี่ยมผืนผ้าที่กำหนด</p>                                                                     |

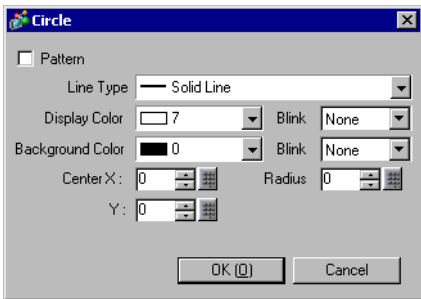
### 21.2.1 Call Screen

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                            |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ฟังก์ชันนี้ใช้เรียกรายการในไลบรารีที่ลงทะเบียนไว้ก่อนหน้านี้ให้แสดงขึ้นมา หน้าจอที่กำหนด (หน้าจอหลัก) จะถูกเรียกให้แสดงขึ้นที่พิกัด X,Y ที่ระบุ ฟังก์ชันนี้ใช้ใน Extended Script ไม่ได้                                                                                                                             |
| รูปแบบ     | <p>b_call (หมายเลขหน้าจอ, พิกัด X, พิกัด Y)</p> <div style="text-align: center;">  </div> <p><b>หมายเหตุ</b></p> <ul style="list-style-type: none"> <li>กำหนดพิกัดกึ่งกลางด้วยพิกัด X และพิกัด Y ของหน้าจอที่ถูกเรียก</li> </ul> |

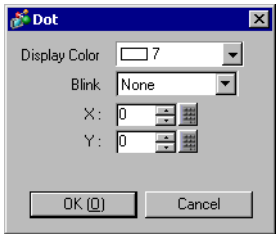
## ตำแหน่งพิกัด



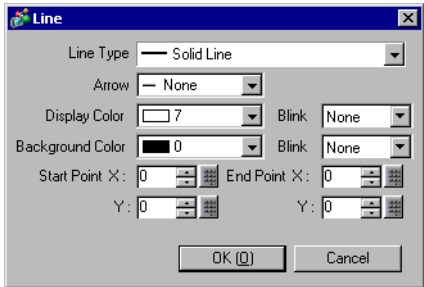
## 21.2.2 Circle

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                     |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | วาดวงกลมตรงตำแหน่งที่กำหนด หากคุณทำเครื่องหมายที่ช่อง [Pattern] จะเป็นการวาดวงกลมทึบ ให้เลือกและป้อนชนิดของเส้น (หรือรูปแบบการเติมเมื่อเลือกรูปแบบ), คุณสมบัติของสี, พิกัดกึ่งกลาง และรัศมี ทั้งยังสามารถตั้งค่าพิกัดกึ่งกลางและรัศมีได้โดยทางอ้อมอีกด้วย                                                                                                    |
| รูปแบบ     | <p>dsp_circle (พิกัด X, พิกัด Y, รัศมี, การกะพริบของสีที่แสดงผล + สีที่แสดงผล, การกะพริบของสีพื้นหลัง + สีพื้นหลัง, ชนิดของเส้น)</p>  <p><b>หมายเหตุ</b></p> <ul style="list-style-type: none"> <li>เมื่อตั้งค่าการกะพริบและสีค่าพร้อมกัน สีพื้นหลังจะโปร่งใส</li> </ul> |

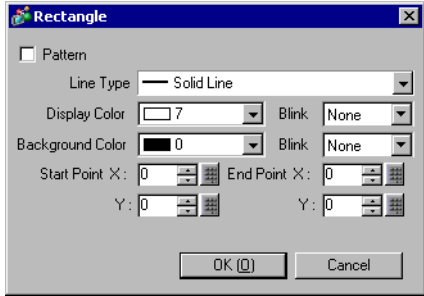
## 21.2.3 Dot

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                        |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | วาดจุดตรงตำแหน่งที่กำหนด กำหนดพิกัด X,Y และสีที่แสดงผล                                                                                                                                                                                                                          |
| รูปแบบ     | <p>dsp_dot (พิกัด X, พิกัด Y, การกะพริบ + สีที่แสดงผล)</p>  <p><b>หมายเหตุ</b></p> <ul style="list-style-type: none"> <li>เมื่อตั้งค่าการกะพริบและสีค่าพร้อมกัน สีพื้นหลังจะโปร่งใส</li> </ul> |

## 21.2.4 Line

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | วาดเส้นตรงตำแหน่งที่กำหนด กำหนดชนิดของเส้น, คุณสมบัติของสี พิกัดเริ่มต้นและพิกัดสิ้นสุด                                                                                                                                                                                                                                                                                                                                                  |
| รูปแบบ     | <p>dsp_line (พิกัด X ของจุดเริ่มต้น, พิกัด Y ของจุดเริ่มต้น, พิกัด X ของจุดสิ้นสุด, พิกัด Y ของจุดสิ้นสุด, การกะพริบของสีที่แสดงผล + สีที่แสดงผล, การกะพริบของสีพื้นหลัง + สีพื้นหลัง, ชนิดของเส้น และลูกศร)</p>  <p><b>หมายเหตุ</b></p> <ul style="list-style-type: none"> <li>เมื่อตั้งค่าการกะพริบและสีค่าพร้อมกัน สีพื้นหลังจะโปร่งใส</li> </ul> |

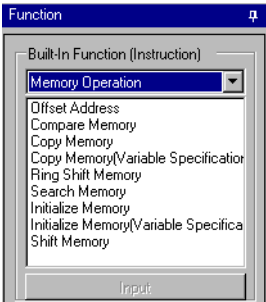
## 21.2.5 Rectangle

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | วาดสี่เหลี่ยมผืนผ้าตรงตำแหน่งที่กำหนด เมื่อคุณทำเครื่องหมายที่ช่อง [Pattern] จะเป็นการวาดสี่เหลี่ยมผืนผ้าทึบ<br>ให้เลือกและป้อนชนิดของเส้น (หรือรูปแบบการเติมเมื่อเลือกรูปแบบ), คุณสมบัติของสี<br>พิกัดเริ่มต้นและพิกัดสิ้นสุด                                                                                                                                                                                                             |
| รูปแบบ     | <p>dsp_rectangle (พิกัด X ของจุดเริ่มต้น, พิกัด Y ของจุดเริ่มต้น, พิกัด X ของจุดสิ้นสุด, พิกัด Y ของจุดสิ้นสุด, การกะพริบของสีที่แสดงผล + สีที่แสดงผล, การกะพริบของสีพื้นหลัง + สีพื้นหลัง, รูปแบบและชนิดของเส้น)</p>  <p><b>หมายเหตุ</b></p> <ul style="list-style-type: none"> <li>เมื่อตั้งค่าการกะพริบและสีตาพร้อมกัน สีพื้นหลังจะโปร่งใส</li> </ul> |

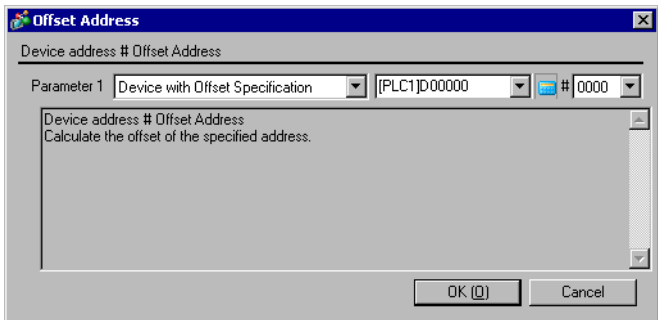
**ข้อสำคัญ**

- เมื่อกำหนดรหัสสีในฟังก์ชันการวาด ให้ตั้งค่ารหัสสีระหว่าง 0 ถึง 255 หากคุณตั้งค่าเป็น E1 ถึง E12 และบันทึกสคริปต์ จะเกิดข้อผิดพลาดขึ้น

## 21.3 การทำงานของหน่วยความจำ

| Memory Operation                                                                  | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Offset Address<br>☞ “21.3.1 Offset Address” (หน้า 21-8)<br>กำหนดค่าออฟเซตของตำแหน่ง                                                                                                                                                                   |
|                                                                                   | Compare Memory<br>☞ “21.3.2 Compare Memory” (หน้า 21-9)<br>เปรียบเทียบข้อมูลสองบล็อตรงตำแหน่งที่ระบุ (ออฟเซต) และเขียนผลการเปรียบเทียบลงในตำแหน่งจัดเก็บ                                                                                              |
|                                                                                   | Copy Memory<br>☞ “21.3.3 Copy Memory” (หน้า 21-12)<br>คัดลอกหน่วยความจำของอุปกรณ์โดยทำเพียงครั้งเดียว                                                                                                                                                 |
|                                                                                   | Copy Memory (Variable Specification)<br>☞ “21.3.4 Copy Memory (Variable Specification)” (หน้า 21-15)<br>คัดลอกหน่วยความจำของอุปกรณ์โดยทำเพียงครั้งเดียว คุณสามารถแก้ไขตำแหน่งต้นทาง (ที่ถูกคัดลอก), ตำแหน่งปลายทาง (ที่คัดลอกแล้ว) และจำนวนตำแหน่งได้ |
|                                                                                   | Ring Shift Memory<br>☞ “21.3.5 Ring Shift Memory” (หน้า 21-16)<br>เลื่อนข้อมูลในหน่วยความจำเป็นวงตามจำนวนเวิร์ดบล็อคที่กำหนด                                                                                                                          |
|                                                                                   | Search Memory<br>☞ “21.3.6 Search Memory” (หน้า 21-19)<br>ค้นหาข้อมูลในบล็อกต่างๆ และส่ง (บันทึก) ผลการค้นหาไปที่ตำแหน่งจัดเก็บข้อมูลที่ระบุไว้                                                                                                       |
|                                                                                   | Initialize Memory<br>☞ “21.3.7 Initialize Memory” (หน้า 21-22)<br>เริ่มการทำงานของอุปกรณ์ทั้งหมดพร้อมกัน                                                                                                                                              |
|                                                                                   | Initialize Memory (Variable Specification)<br>☞ “21.3.8 Initialize Memory (Variable Specification)” (หน้า 21-23)<br>เริ่มการทำงานของอุปกรณ์ทั้งหมดพร้อมกัน คุณสามารถแก้ไขตำแหน่งเริ่มต้น, ข้อมูลที่ตั้งค่าไว้ และจำนวนตำแหน่งได้                      |
|                                                                                   | Shift Memory<br>☞ “21.3.9 Shift Memory” (หน้า 21-24)<br>เลื่อนบล็อกขึ้นข้างบน                                                                                                                                                                         |

### 21.3.1 Offset Address

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |               |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-----------------|--|-----------|-----------|-------|---|--------|-------|---|---------------|----------|---------|--------|----------|----------------|---------------|-------|---|-------|-------|---|------------|
| ข้อมูลสรุป | ฟังก์ชันนี้ใช้สำหรับกำหนดตำแหน่งออฟเซต โดยตำแหน่งจัดเก็บค่าออฟเซตสามารถกำหนดได้เฉพาะตำแหน่งเวิร์ดชั่วคราวเท่านั้น                                                                                                                                                                                                                                                                                                                                                                  |               |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
| รูปแบบ     | [ตำแหน่งอุปกรณ์] # [ตำแหน่งออฟเซต]                                                                                                                                                                                                                                                                                                                                                                                                                                                 |               |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
|            |                                                                                                                                                                                                                                                                                                                                                                                                  |               |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
|            | ช่วงการป้อนค่าคงที่                                                                                                                                                                                                                                                                                                                                                                                                                                                                |               |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
|            | <table><tr><th rowspan="2">ชนิดข้อมูล</th><th colspan="2">การป้อนค่าคงที่</th></tr><tr><th>ค่าต่ำสุด</th><th>ค่าสูงสุด</th></tr><tr><td>Bin16</td><td>0</td><td>65,535</td></tr><tr><td>Bin32</td><td>0</td><td>4,294,967,295</td></tr><tr><td>Bin16+/-</td><td>-32,768</td><td>32,767</td></tr><tr><td>Bin32+/-</td><td>-2,147,483,648</td><td>2,147,483,647</td></tr><tr><td>BCD16</td><td>0</td><td>9,999</td></tr><tr><td>BCD32</td><td>0</td><td>99,999,999</td></tr></table> | ชนิดข้อมูล    | การป้อนค่าคงที่ |  | ค่าต่ำสุด | ค่าสูงสุด | Bin16 | 0 | 65,535 | Bin32 | 0 | 4,294,967,295 | Bin16+/- | -32,768 | 32,767 | Bin32+/- | -2,147,483,648 | 2,147,483,647 | BCD16 | 0 | 9,999 | BCD32 | 0 | 99,999,999 |
| ชนิดข้อมูล | การป้อนค่าคงที่                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |               |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
|            | ค่าต่ำสุด                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | ค่าสูงสุด     |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
| Bin16      | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 65,535        |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
| Bin32      | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 4,294,967,295 |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
| Bin16+/-   | -32,768                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 32,767        |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
| Bin32+/-   | -2,147,483,648                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 2,147,483,647 |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
| BCD16      | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 9,999         |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |
| BCD32      | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 99,999,999    |                 |  |           |           |       |   |        |       |   |               |          |         |        |          |                |               |       |   |       |       |   |            |

#### ตัวอย่างนิพจน์ 1

[w:[PLC1]D0200]=[w:[PLC1]D0100]#[t:0000]

จากตัวอย่างข้างบน เมื่อค่าของ [t:0000] คือ 2 ค่าที่จัดเก็บไว้ในตำแหน่ง D0102 จะถูกออฟเซตให้ตำแหน่ง D0200

#### ตัวอย่างนิพจน์ 2

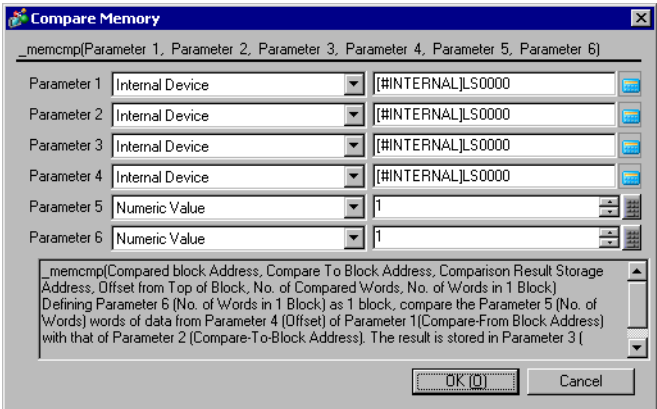
[w:[PLC1]D0100]#[t:0000]=30

จากตัวอย่างข้างบน เมื่อค่าของ [t:0000] คือ 8 จะออฟเซตค่า 30 ให้ตำแหน่ง D0108

#### ข้อสำคัญ

- ตำแหน่งเวิร์ดที่ใช้ในรูปแบบตำแหน่งออฟเซตจะไม่นับเป็นตำแหน่งของ D-Script
- อุปกรณ์ที่เชื่อมต่ออยู่จะไม่อ่านข้อมูลอย่างต่อเนื่องจากอุปกรณ์ที่กำหนดโดยตำแหน่งออฟเซต โดยจะอ่านข้อมูลทุกครั้งที่ใช้งาน D-Script เมื่อเกิดข้อผิดพลาดระหว่างการอ่าน ค่าที่อ่านได้จะถือเป็น "0" นอกจากนี้ บิต 12 ของ LS2032 ซึ่งเป็นรีเลย์พิเศษภายในของเครื่อง GP จะเปิดขึ้นเมื่ออ่านข้อมูลเสร็จสมบูรณ์ตามปกติ บิต 12 จะปิดลง
- หากผลการดำเนินการเกินกว่า 16 บิต (ค่าสูงสุด: 65535) ระบบจะถือว่าบิต 1 ถึงบิต 15 เป็นบิตที่ถูกตัด โดยไม่สนใจบิต 16 และบิตอื่นๆ

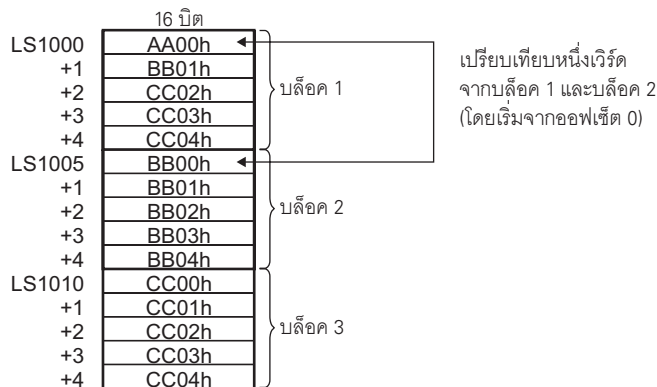
### 21.3.2 Compare Memory

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | <p>เปรียบเทียบข้อมูลสองบล็อกตรงตำแหน่งที่ระบุ (ออฟเซต) และเขียนผลการเปรียบเทียบลงในตำแหน่งจัดเก็บ</p> <p>ระบบจะจัดเก็บผลการเปรียบเทียบด้วยค่าต่อไปนี้ ได้แก่ “0” เมื่อค่าเท่ากัน, 1 เมื่อข้อมูลเป้าหมายมากกว่าข้อมูลเดิม, 2 เมื่อข้อมูลเป้าหมายน้อยกว่าข้อมูลเดิม เมื่อเกิดข้อผิดพลาด ค่าสถานะข้อผิดพลาดจะถูกเขียนลงใน LS9152</p>                                                                                                                                                                                                                                                                                                                                                                           |
| รูปแบบ     | <p>_memcmp ([ตำแหน่งบล็อกที่จะเปรียบเทียบ], [ตำแหน่งบล็อกที่ใช้เปรียบเทียบ], [ตำแหน่งจัดเก็บผลการเปรียบเทียบ], ออฟเซตจากบล็อกส่วนบน, จำนวนเวิร์ดที่จะเปรียบเทียบ, จำนวนเวิร์ดใน 1 บล็อก)</p>  <p>Parameter 1: อุปกรณ์ภายใน</p> <p>Parameter 2: อุปกรณ์ภายใน</p> <p>Parameter 3: อุปกรณ์ภายใน</p> <p>Parameter 4: ค่าตัวเลข (0 ถึง 639), อุปกรณ์ภายใน, ตัวแปรชั่วคราว</p> <p>Parameter 5: ค่าตัวเลข (1 ถึง 640)</p> <p>Parameter 6: ค่าตัวเลข (1 ถึง 640)</p> <p>ข้อมูลที่จะจัดเก็บ</p> <p>0: เหมือนกัน</p> <p>1: ต้นทางน้อยกว่าเป้าหมาย (ต้นทาง &lt; เป้าหมาย)</p> <p>2: ต้นทางมากกว่าเป้าหมาย (ต้นทาง &gt; เป้าหมาย)</p> |

### ตัวอย่างนิพจน์ 1

```
_memcmp ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1005],  
[w:[#INTERNAL]LS0100], 0, 1, 5)
```

(เปรียบเทียบเวิร์ดหนึ่งเวิร์ดจากบล็อก 1 และบล็อก 2 (โดยเริ่มจากออฟเซต 0) และบันทึกผลการเปรียบเทียบใน LS0100)



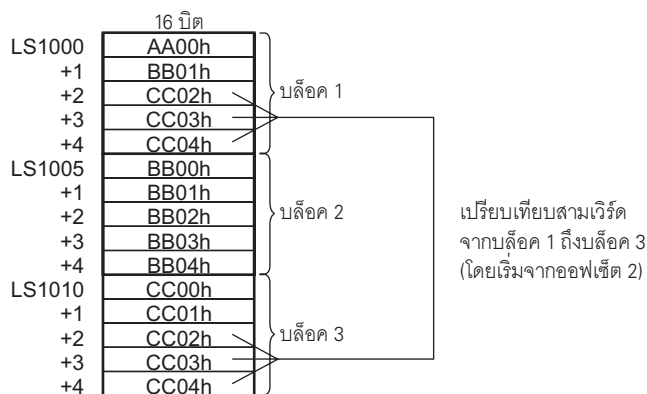
เนื่องจากค่าต้นทางน้อยกว่าค่าเป้าหมาย จึงจัดเก็บผลการเปรียบเทียบ “2” ไว้ใน LS0100



### ตัวอย่างนิพจน์ 2

```
_memcmp ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1010],  
[w:[#INTERNAL]LS0100], 2, 3, 5)
```

(เปรียบเทียบเวิร์ดสามเวิร์ดจากบล็อก 1 และบล็อก 3 (โดยเริ่มจากออฟเซต 2) และบันทึกผลการเปรียบเทียบใน LS0100)



เนื่องจากค่าของข้อมูลต้นทางและข้อมูลเป้าหมายเหมือนกัน จึงจัดเก็บผลการเปรียบเทียบ “0” ไว้ใน LS0100



## สถานะข้อผิดพลาด

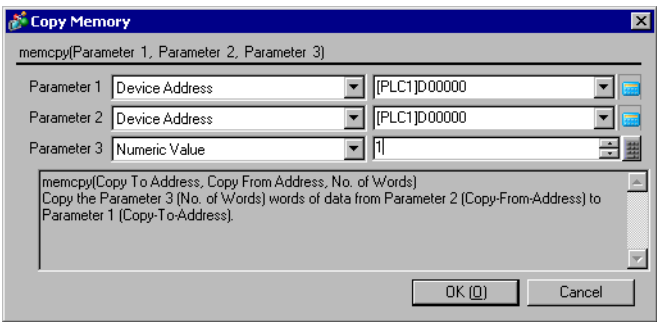
|        |            |
|--------|------------|
| LS9152 | พื้นที่ LS |
|        |            |
|        |            |

| ชื่อฟังก์ชันตัวแก้ไข | พื้นที่ LS | สถานะข้อผิดพลาด | สาเหตุ                         |
|----------------------|------------|-----------------|--------------------------------|
| _memcmp ( )          | LS9152     | 0000h           | เสร็จสมบูรณ์                   |
|                      |            | 0001h           | พารามิเตอร์เกิดข้อผิดพลาด      |
|                      |            | 0003h           | การเขียน/การอ่านเกิดข้อผิดพลาด |

### ข้อสำคัญ

- ช่วงอุปกรณ์ LS ที่สามารถระบุได้นั้นมีเฉพาะอุปกรณ์ในพื้นที่สำหรับผู้ใช้ที่กำหนดเท่านั้น (LS20 ถึง LS2031 และ LS2096 ถึง LS8191)
- หากกำหนดค่าออฟเซตของตอนต้นบล็อกด้วยค่าที่มากกว่าจำนวนเวิร์ดในหนึ่งบล็อก คุณสมบัตินี้จะทำงานไม่ถูกต้อง
- หากกำหนดค่าที่มากกว่าจำนวนเวิร์ดที่จะใช้เปรียบเทียบหรือจำนวนเวิร์ดในหนึ่งบล็อก คุณสมบัตินี้จะทำงานไม่ถูกต้อง

### 21.3.3 Copy Memory

| รายการ     | คำอธิบาย                                                                                                                                                                                                     |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | คัดลอกหน่วยความจำของอุปกรณ์โดยทำเพียงครั้งเดียว ระบบจะคัดลอกข้อมูลจำนวนตำแหน่งไปยังตำแหน่งเวิร์ดของปลายทางการคัดลอก โดยเริ่มจากตำแหน่งเวิร์ดแรกของข้อมูลต้นทาง คุณสามารถป้อนจำนวนตำแหน่งได้ตั้งแต่ 1 ถึง 640 |
| รูปแบบ     | <p>memcpy ([ตำแหน่งปลายทางการคัดลอก], [ตำแหน่งที่ถูกลบ], จำนวนเวิร์ด)</p>                                                  |

#### ตัวอย่างนิพจน์

memcpy ([w:[PLC1]D0200], [w:[PLC1]D0100], 10)

จากตัวอย่างข้างบน ข้อมูลถูกลบออกจากตำแหน่ง D0100-D0109 ไปยังตำแหน่ง D0200-D0209

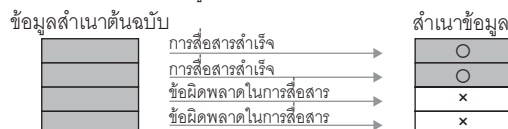
#### ข้อสำคัญ

- เมื่อจำเป็น ข้อมูลสำเนาต้นฉบับจะถูกอ่านจากอุปกรณ์ที่เชื่อมต่อเพียงครั้งเดียวเท่านั้น หากเกิดข้อผิดพลาดในการสื่อสารระหว่างการอ่านข้อมูล บิต 12 ของ LS2032 ซึ่งเป็นรีเลย์พิเศษภายในของ GP จะเปิดขึ้น เมื่ออ่านข้อมูลเสร็จสมบูรณ์ตามปกติ บิต 12 จะปิดลง
- การอ่านจากข้อมูลสำเนาต้นฉบับและการเขียนข้อมูลไปยังปลายทางจะทำเพียงครั้งเดียว หรือทำโดยแบ่งข้อมูลออกเป็นส่วนๆ ให้เท่ากับจำนวนตำแหน่งที่ใช้สำหรับข้อมูลสำเนาต้นฉบับ หากเกิดข้อผิดพลาดในการสื่อสารระหว่างการอ่านข้อมูล ผลการคัดลอกข้อมูลจะแตกต่างกันดังต่อไปนี้ โดยขึ้นอยู่กับว่าจะประมวลผลข้อมูลโดยทำเพียงครั้งเดียวหรือแบ่งข้อมูลเป็นส่วนๆ (ผลการคัดลอกข้อมูล ○: คัดลอกถูกต้อง, ×: ไม่มีการคัดลอกข้อมูล)

(คัดลอกโดยทำเพียงครั้งเดียว)



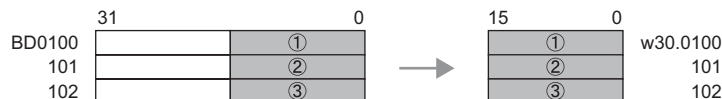
(คัดลอกโดยการแบ่งข้อมูล)



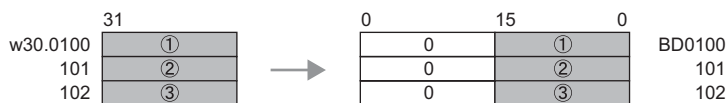
ต่อ

**ข้อสำคัญ**

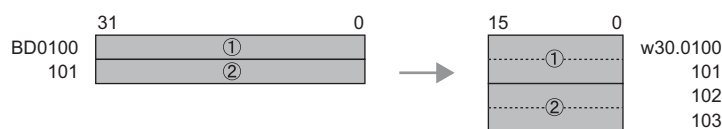
- เมื่อจำนวนตำแหน่งเพิ่มขึ้น การเขียนข้อมูลลงใน PLC จำเป็นต้องใช้เวลานานขึ้น โดยอาจใช้เวลาตั้งแต่ 20 วินาทีจนถึงหลายนาที ขึ้นอยู่กับจำนวนของตำแหน่ง
- หากข้อมูลที่เขียนมีมากกว่าช่วงอุปกรณ์ที่กำหนดไว้ จะเกิดข้อผิดพลาดในการสื่อสาร ในกรณีนี้คุณต้องปิดเครื่อง GP แล้วเปิดอีกครั้งเพื่อตั้งค่า GP ใหม่หลังจากเกิดข้อผิดพลาด
- เมื่อเขียนข้อมูลลงในพื้นที่ LS ด้วยฟังก์ชัน Copy Memory (memcpy) จะสามารถเขียนข้อมูลได้เฉพาะในพื้นที่สำหรับผู้ใช้นั้นๆ คุณไม่สามารถเขียนข้อมูลลงในพื้นที่เก็บข้อมูลระบบ (LS0000 ถึง LS0019), พื้นที่พิเศษ (LS2032 ถึง LS2047) หรือพื้นที่สำรอง (LS2048 ถึง LS2095) แต่สามารถอ่านข้อมูลจากพื้นที่เหล่านี้ได้
- เมื่อคัดลอกข้อมูลของอุปกรณ์ชนิด 32 บิตไปยังอุปกรณ์ชนิด 16 บิตโดยใช้ D-Script และกำหนดความยาวบิตเป็น 16 บิต ระบบจะคัดลอกเฉพาะข้อมูลของ 16 บิตล่างเท่านั้น
- ตัวอย่าง memcpy ([w:[PLC1]w30.0100], [w:[PLC1]BD0100], 3)



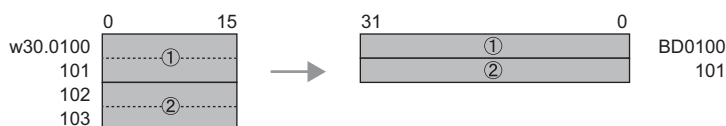
- นอกจากนี้ เมื่อคัดลอกข้อมูลของอุปกรณ์ชนิด 16 บิตไปยังอุปกรณ์ชนิด 32 บิต ระบบจะคัดลอกเฉพาะข้อมูลของ 16 บิตล่างและกำหนดค่า “0” ให้กับข้อมูลของ 16 บิตบน
- เช่น memcpy ([w:[PLC1]BD0100], [w:[PLC1]w30.0100], 3)



- เมื่อคัดลอกข้อมูลของอุปกรณ์ชนิด 32 บิตไปยังอุปกรณ์ชนิด 16 บิต หรือเมื่อคัดลอกข้อมูลของอุปกรณ์ชนิด 16 บิตไปยังอุปกรณ์ชนิด 32 บิต หากกำหนดความยาวบิตของ D-Script ไว้เท่ากับ 32 การคัดลอกจะเป็นดังต่อไปนี้ เมื่ออุปกรณ์หนึ่งเป็นอุปกรณ์ชนิด 32 บิต และอีกชิ้นหนึ่งเป็นอุปกรณ์ชนิด 16 บิต ให้ใช้จำนวนตำแหน่งของอุปกรณ์ชนิด 16 บิต เพื่อกำหนดจำนวนตำแหน่งของฟังก์ชัน memcpy ( )
- ตัวอย่าง memcpy ([w:[PLC1]w30.0100], [w:[PLC1]BD0100], 4)

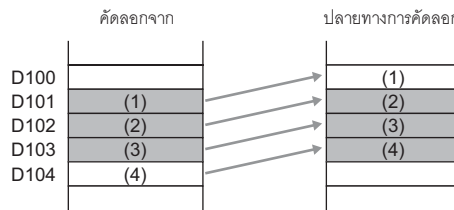


- ตัวอย่าง memcpy ([w:[PLC1]BD0100], [w:[PLC1]w30.0100], 4)

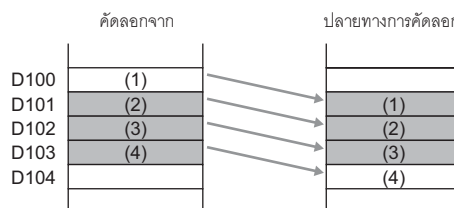


**ข้อสำคัญ**

- หากข้อมูลต้นฉบับและข้อมูลปลายทางมีช่วงข้อมูลคาบเกี่ยวกัน ข้อมูลที่คาบเกี่ยวกันทั้งหมด จะถูกเขียนใหม่ดังนี้
- ตัวอย่าง เมื่อคัดลอก D101-D104 ไปยัง D100-D103
- ข้อมูลถูกคัดลอกลงในตำแหน่งที่มีหมายเลขน้อยกว่า

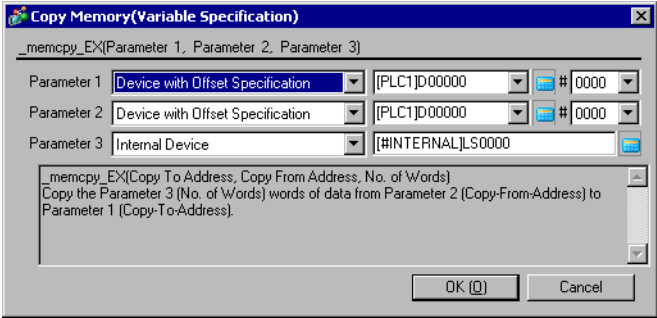


- ตัวอย่าง เมื่อคัดลอก D100-D103 ไปยัง D101-D104
- ข้อมูลถูกคัดลอกลงในตำแหน่งที่มีหมายเลขมากกว่า



- แม้ว่าฟังก์ชันของตัวอย่างนี้จะกำหนดไว้ 2 ตำแหน่ง แต่ตำแหน่งดังกล่าวจะไม่นับเป็นตำแหน่งของ D-Script
- เมื่อใช้ตำแหน่งอุปกรณ์กับคำสั่ง Assign ค่าที่เขียนจะไม่ถูกกำหนดในทันที เนื่องจากต้องใช้เวลาในการสื่อสารกับอุปกรณ์/PLC เมื่อกำหนดตำแหน่งอุปกรณ์ไว้ ระบบจะไม่กำหนดค่าที่เขียนให้โดยทันที เนื่องจากต้องใช้เวลาในการสื่อสารกับอุปกรณ์/PLC

### 21.3.4 Copy Memory (Variable Specification)

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                           |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | คัดลอกหน่วยความจำของอุปกรณ์โดยทำเพียงครั้งเดียว ข้อมูลของตำแหน่งที่ระบุใน Parameter 3 จะถูกคัดลอกจากตำแหน่งเวิร์ดต้นทาง (ที่ถูกคัดลอก) ที่ระบุใน Parameter 2 ไปยังตำแหน่งเวิร์ดปลายทาง (ปลายทางการคัดลอก) ที่ระบุใน Parameter 1<br>คุณสามารถป้อนจำนวนตำแหน่งได้ตั้งแต่ 1 ถึง 640 ฟังก์ชัน “_memcpy_EX” ช่วยให้กำหนดตำแหน่งต้นทาง ตำแหน่งปลายทาง และจำนวนตำแหน่งโดยทางอ้อมได้                       |
| รูปแบบ     | <p>_memcpy_EX ([ตำแหน่งปลายทางการคัดลอก], [ตำแหน่งที่ถูกคัดลอก], จำนวนเวิร์ด)</p> <p>Parameter 1: ตำแหน่งอุปกรณ์ + ตำแหน่งชั่วคราว</p> <p>Parameter 2: ตำแหน่งอุปกรณ์ + ตำแหน่งชั่วคราว</p> <p>Parameter 3: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 3 คือตั้งแต่ 1 ถึง 640)</p>  |

#### ตัวอย่างนิพจน์

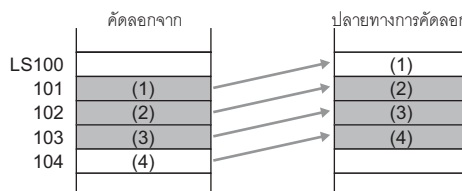
[t:0000]=10, [t:0001]=20

\_memcpy\_EX ([w:[#INTERNAL]LS0100]#[t:0000], [w:[PLC1]D0100]#[t:0001], 5)

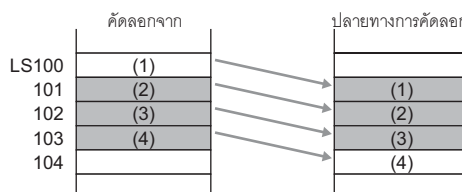
จากตัวอย่างข้างบน ระบบจะอ่านข้อมูลจำนวนห้าเวิร์ดจากตำแหน่ง D0120 และเขียนลงใน LS0110 ถึง LS0114

#### ข้อสำคัญ

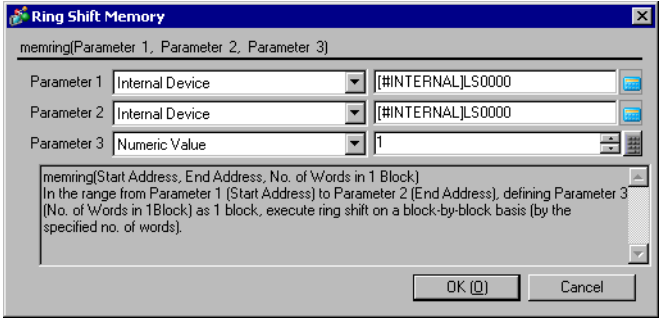
- หากข้อมูลต้นฉบับและข้อมูลปลายทางมีช่วงข้อมูลคาบเกี่ยวกัน ข้อมูลที่คาบเกี่ยวกันทั้งหมด จะถูกเขียนใหม่ดังนี้
- ตัวอย่าง เมื่อคัดลอกจาก LS101-LS104 ไปยัง LS100-LS103
- ข้อมูลถูกคัดลอกลงในตำแหน่งที่มีหมายเลขน้อยกว่า



- ตัวอย่าง เมื่อคัดลอกจาก LS100-LS103 ไปยัง LS101-LS104
- ข้อมูลถูกคัดลอกลงในตำแหน่งที่มีหมายเลขมากกว่า

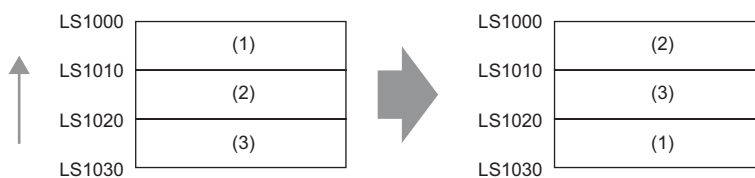


### 21.3.5 Ring Shift Memory

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | <p>เลื่อนข้อมูลในหน่วยความจำในบล็อกให้เป็นวง<br/>ฟังก์ชันนี้จะเลื่อนข้อมูลให้เป็นวงระหว่างตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดในบล็อก<br/>(ตามจำนวนเวิร์ดที่กำหนด) เมื่อเกิดข้อผิดพลาด สถานะข้อผิดพลาดจะถูกเขียนลงใน LS9150</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| รูปแบบ     | <p>memring ([ตำแหน่งเริ่มต้น], [ตำแหน่งสิ้นสุด], จำนวนเวิร์ดใน 1 บล็อก)</p>  <p>Parameter 1: อุปกรณ์ภายใน<br/>Parameter 2: อุปกรณ์ภายใน<br/>Parameter 3: ค่าตัวเลข (1 ถึง 640)</p> <ul style="list-style-type: none"> <li>เมื่อ Parameter 1 น้อยกว่า Parameter 2 (<math>P1 &lt; P2</math>) ข้อมูลบล็อกจะเลื่อนขึ้นข้างบน</li> <li>เมื่อ Parameter 1 มากกว่า Parameter 2 (<math>P1 &gt; P2</math>) ข้อมูลบล็อกจะเลื่อนลงข้างล่าง</li> </ul> <p><b>ข้อสำคัญ</b></p> <ul style="list-style-type: none"> <li>โปรดตรวจสอบให้แน่ใจว่าได้ตั้งค่าตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดเป็นอุปกรณ์ชนิดเดียวกัน (LS หรือ USR)</li> </ul> |

#### ตัวอย่างนิพจน์ 1

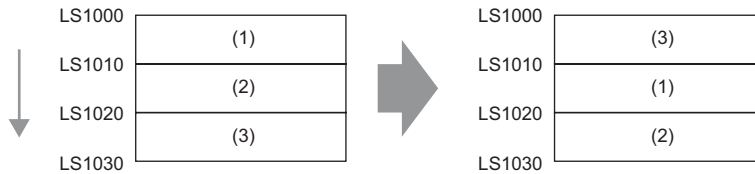
memring ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1030], 10)  
(เมื่อ Parameter 1 น้อยกว่า Parameter 2 ( $P1 < P2$ ))



ข้อมูลจะเลื่อนขึ้นข้างบน 10 เวิร์ดบล็อก

## ตัวอย่างนิพจน์ 2

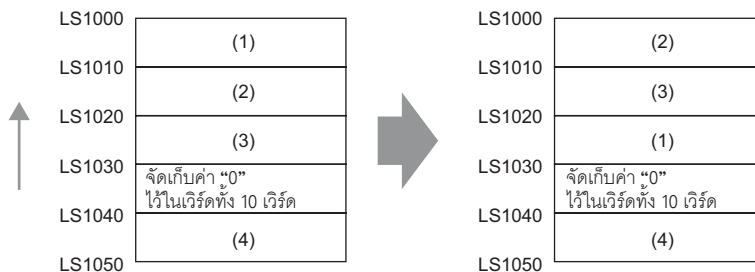
memring ([w:[#INTERNAL]LS1030], [w:[#INTERNAL]LS1000], 10)  
(เมื่อ Parameter 1 มากกว่า Parameter 2 ( $P1 > P2$ ))



ข้อมูลจะเลื่อนลงข้างล่าง 10 เวิร์ดบล็อก

## ตัวอย่างนิพจน์ 3

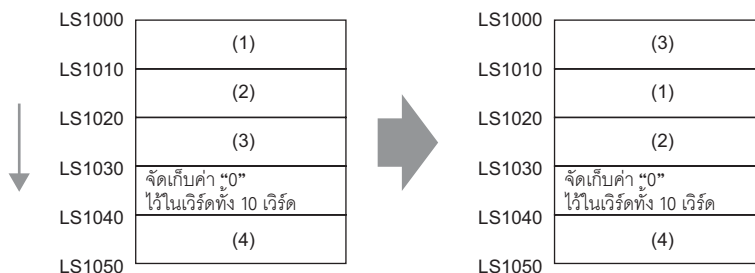
memring ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1050], 10)  
(เมื่อมีบล็อกที่มีเวิร์ดทุกเวิร์ดมีข้อมูลเป็น “0” อยู่ภายในช่วง)



ข้อมูลจะเลื่อนขึ้นข้างบน 10 เวิร์ดบล็อกเท่านั้น โดยเริ่มจากบล็อกเริ่มต้นจนถึงบล็อกที่มีข้อมูลเป็น “0” หากมีข้อมูลอยู่ต่อจากบล็อกที่มีข้อมูลเป็น “0” ระบบจะไม่สนใจข้อมูลนั้น

## ตัวอย่างนิพจน์ 4

memring ([w:[#INTERNAL]LS1050], [w:[#INTERNAL]LS1000], 10)  
(เมื่อมีบล็อกที่มีข้อมูลเป็น “0” อยู่ภายในช่วง)



ข้อมูลจะเลื่อนลงข้างล่าง 10 เวิร์ดบล็อกเท่านั้น โดยเริ่มจากบล็อกเริ่มต้นจนถึงบล็อกที่มีข้อมูลเป็น “0” หากมีข้อมูลอยู่ต่อจากบล็อกที่มีข้อมูลเป็น “0” ระบบจะไม่สนใจข้อมูลนั้น

## สถานะข้อผิดพลาด

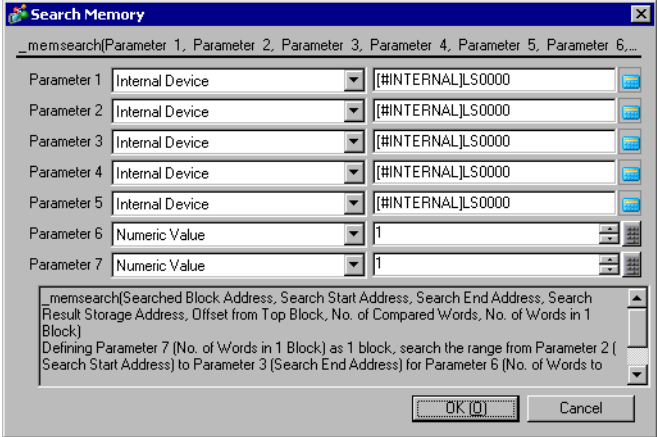
|        |            |
|--------|------------|
| LS9150 | พื้นที่ LS |
|        |            |
|        |            |

| ชื่อฟังก์ชันตัวแก้ไข | พื้นที่ LS | สถานะข้อผิดพลาด | สาเหตุ                         |
|----------------------|------------|-----------------|--------------------------------|
| memring ( )          | LS9150     | 0000h           | เสร็จสมบูรณ์                   |
|                      |            | 0001h           | พารามิเตอร์เกิดข้อผิดพลาด      |
|                      |            | 0003h           | การเขียน/การอ่านเกิดข้อผิดพลาด |

### ข้อสำคัญ

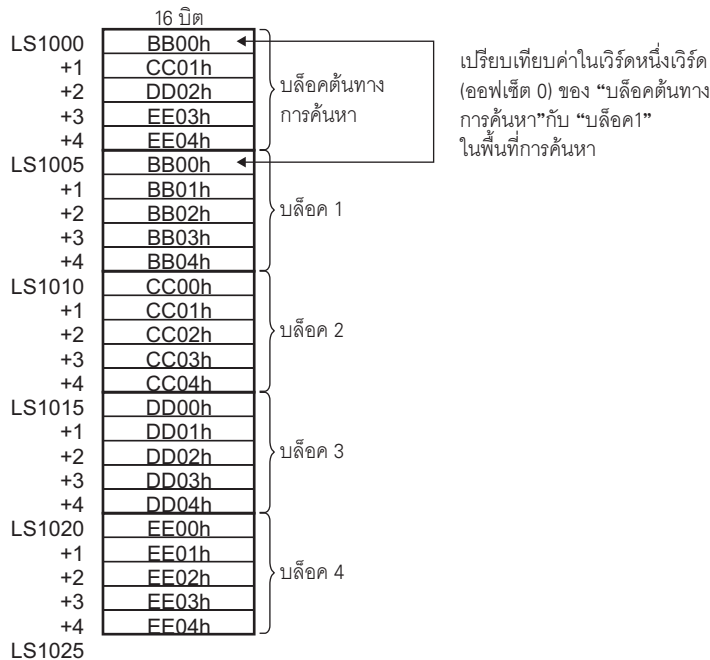
- เวลาที่ต้องใช้ในการประมวลผลจะเป็นสัดส่วนกับช่วงที่กำหนดโดยตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุด ยิ่งกำหนดช่วงไวกว ยิ่งต้องใช้เวลาในการประมวลผลนานขึ้น ระบบจะไม่รีเฟรชพาร์ทจนกว่าจะประมวลผลเสร็จสมบูรณ์
- ช่วงอุปกรณ์ LS ที่สามารถระบุได้นั้นมีเฉพาะอุปกรณ์ในพื้นที่สำหรับผู้ใช้ที่กำหนดเท่านั้น (LS20 ถึง LS2031 และ LS2096 ถึง LS8191)

### 21.3.6 Search Memory

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ฟังก์ชันนี้จะค้นหาข้อมูลในบล็อกโดยเริ่มจากรายการแรกในช่วงที่ระบุ โดยการเปรียบเทียบบล็อกข้อมูล โดยเริ่มจากบล็อก (ออฟเซต) ที่ระบุ และส่ง (บันทึก) ผลการค้นหาไปที่ตำแหน่งจัดเก็บข้อมูลที่ระบุไว้ เมื่อพบบล็อกที่ตรงกัน จะบันทึกค่าออฟเซตของบล็อกไว้ (1 หรือมากกว่า) เมื่อไม่พบบล็อกที่ตรงกัน จะบันทึก “FFFFh” ไว้ เมื่อเกิดข้อผิดพลาด ค่าสถานะข้อผิดพลาดจะถูกเขียนลงใน LS9153                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| รูปแบบ     | <p>_memsearch ([ตำแหน่งบล็อกที่ค้นหา], [ตำแหน่งเริ่มต้นการค้นหา], [ตำแหน่งสิ้นสุดการค้นหา], [ตำแหน่งจัดเก็บผลการค้นหา], ออฟเซตจากบล็อกส่วนบน, จำนวนเวิร์ดที่จะเปรียบเทียบ, จำนวนเวิร์ดใน 1 บล็อก)</p>  <p>Parameter 1: อุปกรณ์ภายใน</p> <p>Parameter 2: อุปกรณ์ภายใน</p> <p>Parameter 3: อุปกรณ์ภายใน</p> <p>Parameter 4: อุปกรณ์ภายใน</p> <p>Parameter 5: ค่าตัวเลข (0 ถึง 639), อุปกรณ์ภายใน, ตัวแปรชั่วคราว</p> <p>Parameter 6: ค่าตัวเลข (1 ถึง 640)</p> <p>Parameter 7: ค่าตัวเลข (1 ถึง 640)</p> <p>ข้อมูลที่จะถูกเขียน</p> <p>เมื่อมีบล็อกที่ตรงกัน: ค่าออฟเซตของบล็อก (“1” หรือสูงกว่า)</p> <p>เมื่อไม่มีบล็อกที่ตรงกัน: “FFFFh”</p> <p><b>ข้อสำคัญ</b></p> <ul style="list-style-type: none"> <li>โปรดตรวจสอบให้แน่ใจว่าได้ตั้งค่าตำแหน่งเริ่มต้นการค้นหาและตำแหน่งสิ้นสุดการค้นหาเป็นอุปกรณ์ชนิดเดียวกัน (LS หรือ USR) อย่างไรก็ตาม สามารถตั้งค่า [Searched Block Address] และ [Search Result Storage Address] ให้เป็นอุปกรณ์ภายในได้</li> <li>ค่า [Parameter 2] ต้องน้อยกว่า [Parameter 3] (Parameter 2 &lt; Parameter 3) มิฉะนั้นจะเกิดข้อผิดพลาดขึ้น</li> </ul> |

### ตัวอย่างนิพจน์ 1

\_memsearch ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1005],  
[w:[#INTERNAL]LS1025], [w:[#INTERNAL]LS0100], 0, 1, 5)  
(ค้นหาระหว่าง LS1005 ถึง LS1025 เพื่อหาบล็อกที่มีค่าเดียวกัน โดยเริ่มต้นจากออฟเซต 0 ของบล็อกต้นทาง  
การค้นหา และจัดเก็บผลการค้นหาไว้ใน LS0100)

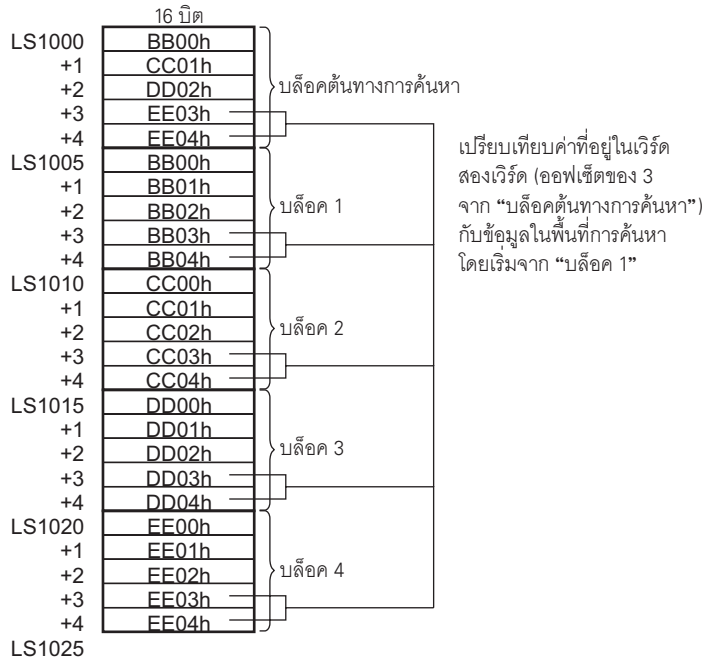


ในกรณีนี้ ค่าของ “บล็อก 1” ตรงกับค่าของ “บล็อกต้นทางการค้นหา” ดังนั้น จะจัดเก็บผลการค้นหา “1”  
ไว้ใน LS0100

|        |   |
|--------|---|
| LS0100 |   |
|        | 1 |
|        |   |

### ตัวอย่างนิพจน์ 2

\_memsearch ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1005],  
[w:[#INTERNAL]LS1025], [w:[#INTERNAL]LS0100], 3, 2, 5)  
(ค้นหาระหว่าง LS1005 ถึง LS1025 เพื่อหาบล็อกที่มีค่าเดียวกัน โดยใช้เวิร์ดสองเวิร์ด โดยเริ่มค้นหาจาก  
ออฟเซตของ 3 และจัดเก็บผลลัพธ์ไว้ใน LS0100)



ในกรณีนี้ ค่าของ “บล็อก 4” จะตรงกับค่าของ “บล็อกต้นทางการค้นหา” ดังนั้น จะจัดเก็บผลการค้นหา “4” ไว้ใน LS0100

|        |   |
|--------|---|
| LS0100 | 4 |
|--------|---|

### สถานะข้อผิดพลาด

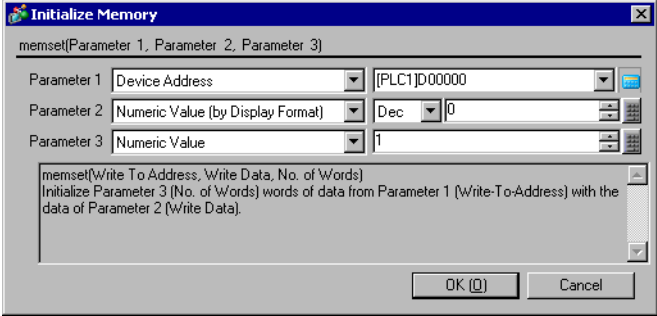
|        |            |
|--------|------------|
| LS9153 | พื้นที่ LS |
|--------|------------|

| ชื่อฟังก์ชันตัวแก้ไข | พื้นที่ LS | สถานะข้อผิดพลาด | สาเหตุ                         |
|----------------------|------------|-----------------|--------------------------------|
| _memsearch ( )       | LS9153     | 0000h           | เสร็จสมบูรณ์                   |
|                      |            | 0001h           | พารามิเตอร์เกิดข้อผิดพลาด      |
|                      |            | 0003h           | การเขียน/การอ่านเกิดข้อผิดพลาด |

#### ข้อสำคัญ

- เวลาที่ต้องใช้ในการประมวลผลจะเป็นสัดส่วนกับช่วงที่กำหนดโดยตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุด ยิ่งกำหนดช่วงไว้นาน ยิ่งต้องใช้เวลาในการประมวลผลนานขึ้น ระบบจะไม่รีเฟรชพารามิเตอร์จนกว่าจะประมวลผลเสร็จสมบูรณ์
- ช่วงอุปกรณ์ LS ที่สามารถระบุได้นั้นมีเฉพาะอุปกรณ์ในพื้นที่สำหรับผู้ใช้ที่กำหนดเท่านั้น (LS20 ถึง LS2031 และ LS2096 ถึง LS8191)

### 21.3.7 Initialize Memory

| รายการ     | คำอธิบาย                                                                                                                                                 |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | เริ่มการทำงานของอุปกรณ์ทั้งหมดพร้อมกัน ข้อมูลการตั้งค่าจำนวนตำแหน่งจะได้อมาจากตำแหน่งเวิร์ดที่ตั้งค่า คุณสามารถป้อนจำนวนตำแหน่งได้ตั้งแต่ 1 ถึง 640      |
| รูปแบบ     | <p>memset ([ตำแหน่งปลายทางการเขียน], ข้อมูลที่เขียน, จำนวนเวิร์ด)</p>  |

#### ตัวอย่างนิพจน์

memset ([w:[PLC1]D0100], 0, 10)

จากตัวอย่างข้างบน เป็นการตั้งค่า “0” ให้กับตำแหน่ง D0100 ถึง D0109

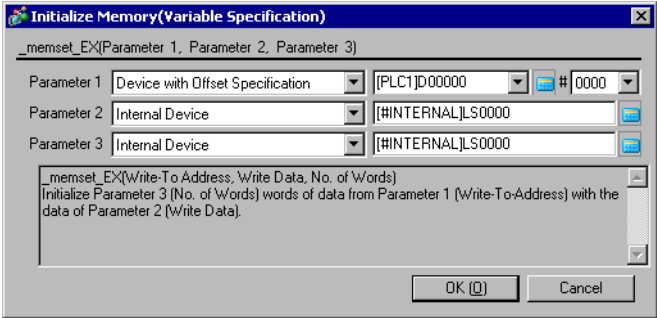
#### ข้อสำคัญ

- เมื่อจำนวนตำแหน่งเพิ่มขึ้น การเขียนข้อมูลลงใน PLC จำเป็นต้องใช้เวลานานขึ้น โดยอาจใช้เวลาตั้งแต่ 20 วินาทีจนถึงหลายนาที ขึ้นอยู่กับจำนวนของตำแหน่ง
- หากข้อมูลที่จะเขียนมีมากกว่าช่วงอุปกรณ์ที่กำหนดไว้ จะเกิดข้อผิดพลาดในการสื่อสาร ในกรณีนี้ คุณต้องปิดเครื่อง GP แล้วเปิดอีกครั้งเพื่อตั้งค่า GP ใหม่หลังจากเกิดข้อผิดพลาด
- แม้ว่าฟังก์ชันนี้จะกำหนดตำแหน่งไว้ แต่ตำแหน่งดังกล่าวจะไม่นับเป็นตำแหน่งของ D-Script
- เมื่อเขียนข้อมูลในพื้นที่ LS ด้วยฟังก์ชัน Memory Reset (memset) จะสามารถเขียนข้อมูลได้เฉพาะในพื้นที่สำหรับผู้ใช้นั้นๆ คุณไม่สามารถเขียนข้อมูลลงในพื้นที่เก็บข้อมูลระบบ (LS0000 ถึง LS0019), พื้นที่พิเศษ (LS2032 ถึง S2047) หรือพื้นที่สำรอง (LS2048 ถึง LS2095)
- เมื่อใช้ตำแหน่งอุปกรณ์กับคำสั่ง Assign ระบบจะไม่กำหนดค่าที่เขียนให้โดยทันที เนื่องจากต้องใช้เวลาในการส่งข้อมูลจาก GP ไปยัง PLC
- ตัวอย่าง
 

```
memset ([w:D0100], 0, 10) // Initializes “D100 to D109” as 0
[w:D200] = [w:D100]      // Assigns D100 data to D200.
```

ในตัวอย่างนี้ ค่า 0 ซึ่งเป็นผลการดำเนินการที่เขียนลงใน D100 นั้นยังไม่ถูกกำหนดลงใน D200

### 21.3.8 Initialize Memory (Variable Specification)

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | เริ่มการทำงานของอุปกรณ์ทั้งหมดพร้อมกัน ข้อมูลตั้งค่าซึ่งระบุด้วย Parameter 2 จะถูกตั้งค่าจากตำแหน่งเวิร์ดที่ตั้งค่าซึ่งระบุด้วย Parameter 1 ลงในตำแหน่งซึ่งระบุด้วย Parameter 3 คุณสามารถป้อนจำนวนตำแหน่งได้ตั้งแต่ 1 ถึง 640 และยังสามารถกำหนดตำแหน่งปลายทางการเขียน, ข้อมูลที่เขียน และหมายเลขตำแหน่งได้โดยอ้อม                                                                                                                                                                                       |
| รูปแบบ     | <p>_memset_EX ([ตำแหน่งปลายทางการเขียน], ข้อมูลที่เขียน, จำนวนเวิร์ด)</p>  <p>Parameter 1: ตำแหน่งอุปกรณ์ + ตำแหน่งชั่วคราว</p> <p>Parameter 2: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 2 คือตั้งแต่ 0 ถึง 65535 สำหรับ Dec และตั้งแต่ 0 ถึง FFFF สำหรับ Hex)</p> <p>Parameter 3: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 3 คือตั้งแต่ 1 ถึง 640)</p> |

#### ตัวอย่างนิพจน์

[t:0000]=10

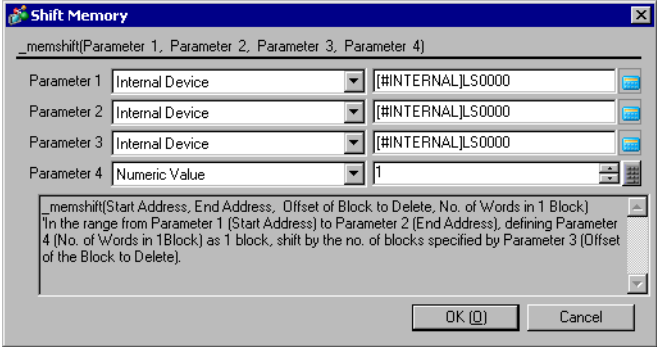
[w:LS0050]=0

[w:LS0051]=5

\_memset\_EX ([w:[#INTERNAL]LS0100]#[t:0000], [w:[#INTERNAL]LS0050],  
[w:[#INTERNAL]LS0051])

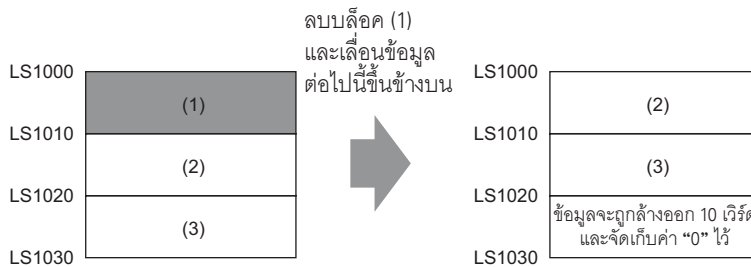
จากตัวอย่างข้างบน ระบบจะเขียนค่า “0” ลงในเวิร์ดหัวเวิร์ดตั้งแต่ LS0100 ถึง LS0114

### 21.3.9 Shift Memory

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ลบบล็อกที่ระบุและเลื่อนบล็อกข้อมูลต่อไปยังขั้นข้างบน บล็อกที่จะลบจะถูกกำหนดโดยใช้ออฟเซตเมื่อเกิดข้อผิดพลาด สถานะข้อผิดพลาดจะถูกเขียนลงใน LS9151                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| รูปแบบ     | <p>_memshift ([ตำแหน่งเริ่มต้น], [ตำแหน่งสิ้นสุด], ออฟเซตของบล็อกที่จะลบ, จำนวนเวิร์ดใน 1 บล็อก)</p>  <p>Parameter 1: อุปกรณ์ภายใน<br/>         Parameter 2: อุปกรณ์ภายใน<br/>         Parameter 3: ค่าตัวเลข (1 ถึง 65,535), อุปกรณ์ภายใน, ตัวแปรชั่วคราว<br/>         Parameter 4: ค่าตัวเลข (1 ถึง 640)</p> <p><b>ข้อสำคัญ</b></p> <ul style="list-style-type: none"> <li>โปรดตรวจสอบให้แน่ใจว่าได้ตั้งค่าตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดเป็นอุปกรณ์ชนิดเดียวกัน (LS หรือ USR)</li> <li>ค่า [Parameter 1] ต้องน้อยกว่า [Parameter 2] (Parameter 1 &lt; Parameter 2) มิฉะนั้นจะเกิดข้อผิดพลาดขึ้น</li> </ul> |

## ตัวอย่างนิพจน์ 1

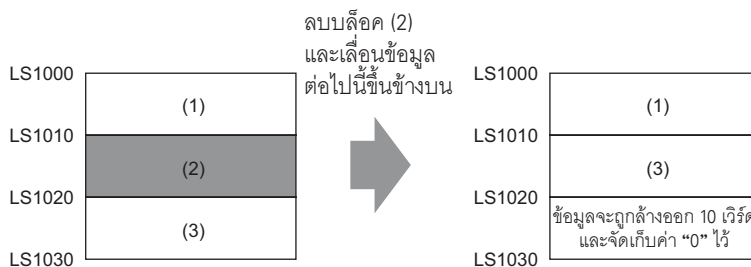
`_memshift ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1030], 1, 10)`



ข้อมูลเลื่อนขึ้นข้างบนเป็นบล็อก (1 บล็อก = 10 เวิร์ด) และบล็อกสุดท้าย (10 เวิร์ด) ถูกล้างข้อมูลเป็นศูนย์

## ตัวอย่างนิพจน์ 2

`_memshift ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1030], 2, 10)`



ข้อมูลเลื่อนขึ้นข้างบนเป็นบล็อก (1 บล็อก = 10 เวิร์ด) โดยเริ่มจากตำแหน่งออฟเซต 2 และบล็อกสุดท้าย (10 เวิร์ด) ถูกล้างข้อมูลเป็นศูนย์

## สถานะข้อผิดพลาด

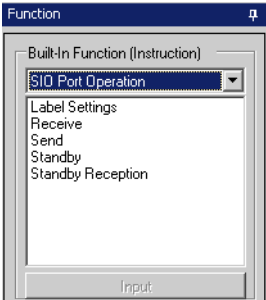
|        |            |
|--------|------------|
|        | พื้นที่ LS |
| LS9151 |            |
|        |            |

| ชื่อฟังก์ชันตัวแก้ไข       | พื้นที่ LS | สถานะข้อผิดพลาด | สาเหตุ                         |
|----------------------------|------------|-----------------|--------------------------------|
| <code>_memshift ( )</code> | LS9151     | 0000h           | เสร็จสมบูรณ์                   |
|                            |            | 0001h           | พารามิเตอร์เกิดข้อผิดพลาด      |
|                            |            | 0003h           | การเขียน/การอ่านเกิดข้อผิดพลาด |

### ข้อสำคัญ

- เวลาที่ใช้ในการประมวลผลจะเป็นสัดส่วนกับช่วงที่กำหนดโดยตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุด ยิ่งกำหนดช่วงไวกว่า ยิ่งต้องใช้เวลาในการประมวลผลนานขึ้น ระบบจะไม่รีเฟรชพาร์ทิชันกว่าจะประมวลผลเสร็จสมบูรณ์
- หากกำหนดค่าออฟเซตของบล็อกที่จะลบด้วยค่าที่เกินกว่าช่วงตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดที่ระบุไว้ คุณสมบัตินี้จะทำงานไม่ถูกต้อง
- ช่วงอุปกรณ์ LS ที่สามารถระบุได้นั้นมีเฉพาะอุปกรณ์ในพื้นที่สำหรับผู้ใช้ที่กำหนดเท่านั้น (LS20 ถึง LS2031 และ LS2096 ถึง LS8191)

## 21.4 การทำงานของพอร์ต SIO

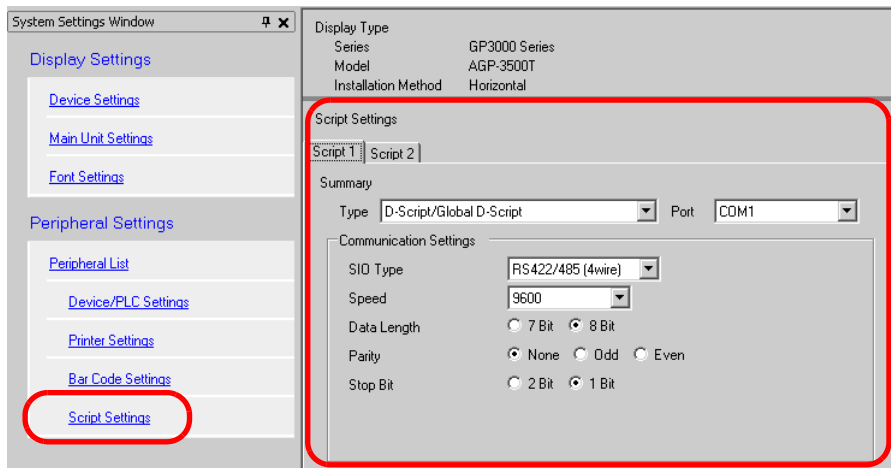
| SIO Port Operation                                                                | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Label Settings</p> <p>☞ “21.4.1 Label Settings” (หน้า 21-28)</p> <p>กำหนดจากตัวแปรควบคุม, สถานะ, จำนวนข้อมูลที่ได้รับ, ฟังก์ชันรับและฟังก์ชันส่ง</p>                                                            |
|                                                                                   | <p>Receive</p> <p>☞ “21.4.2 Receive” (หน้า 21-32)</p> <p>อ่านข้อมูลที่ได้รับเข้ามาจากพอร์ตอนุกรมที่กำหนด (COM1 หรือ COM2)</p>                                                                                      |
|                                                                                   | <p>Send</p> <p>☞ “21.4.3 Send” (หน้า 21-33)</p> <p>เขียนข้อมูลลงในพอร์ตอนุกรมที่กำหนด (COM1 หรือ COM2)</p>                                                                                                         |
|                                                                                   | <p>Extended Receive</p> <p>☞ “21.4.4 Extended Receive” (หน้า 21-34)</p> <p>อ่านข้อมูลที่ได้รับเข้ามาจากพอร์ตอนุกรมที่กำหนด (COM1 หรือ COM2)</p> <p>ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น</p>             |
|                                                                                   | <p>Extended Send</p> <p>☞ “21.4.5 Extended Send” (หน้า 21-35)</p> <p>เขียนข้อมูลลงในพอร์ตอนุกรมที่กำหนด (COM1 หรือ COM2)</p> <p>ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น</p>                                |
|                                                                                   | <p>ฟังก์ชัน Standby Reception</p> <p>☞ “21.4.6 ฟังก์ชัน Standby Reception” (หน้า 21-36)</p> <p>ระบบอยู่ในโหมดสแตนด์บายการรับจนกว่าจะได้รับสตริงที่ระบุ</p> <p>ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น</p>  |
|                                                                                   | <p>ฟังก์ชัน Standby</p> <p>☞ “21.4.7 ฟังก์ชัน Standby” (หน้า 21-37)</p> <p>ระบบหยุดรอ (พักการทำงานชั่วคราว) เป็นระยะเวลาตามที่ระบุไว้จนกว่าจะเริ่มทำงาน</p> <p>ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น</p> |

ข้อสำคัญ

- คุณสามารถใส่ตัวแปร Label Settings, Send และ Receive ไว้ใน D-Script/Global D-Script ได้อย่างง่ายดาย
- หากสื่อสารกับ D-Scripts/Global D-Scripts โปรดตรวจสอบให้แน่ใจว่าได้กำหนดการตั้งค่าสคริปต์ต่อไปนี้แล้ว หากไม่ได้กำหนดการตั้งค่าสคริปต์ จะดำเนินการไม่ได้

ขั้นตอนการตั้งค่า [D-Script/Global D-Script]

- (1) คลิก [Project] - [System Settings] - [Script Settings]  
ตั้งค่า [Type] เป็น “D-Script/Global D-Script”



แท็บสำหรับตั้งค่าสคริปต์มีด้วยกัน 2 แท็บ รูปทางด้านบนนี้ใช้ [Script 1]

ตั้งค่า [Port] เป็น COM1 หรือ COM2 แล้วตั้งค่า [Communication Settings] ให้ตรงกับ Extended SIO

- เมื่อสร้างโปรแกรมการสื่อสารด้วยฟังก์ชันการทำงานที่สูงกว่าการทำงานของพอร์ต SIO ขอแนะนำให้ใช้ [Extended Script] สำหรับตัวอย่างที่ใช้ extended script โปรดดูที่ “20.5 การสื่อสารกับอุปกรณ์ต่อพ่วงที่สคริปต์ปกติไม่รองรับ” (หน้า 20-22)

### 21.4.1 Label Settings

◆ ตัวแปรควบคุม

เมื่อกำหนดบิต: [c:EXT\_SIO\_CTRL\*\*] (เขียนอย่างเดียว)

เมื่อกำหนดเวิร์ด: [c:EXT\_SIO\_CTRL] (เขียนอย่างเดียว)

◆ สถานะ

เมื่อกำหนดบิต: [s:EXT\_SIO\_STAT\*\*] (อ่านอย่างเดียว)

เมื่อกำหนดเวิร์ด: [s:EXT\_SIO\_STAT] (อ่านอย่างเดียว)

◆ จำนวนข้อมูลที่ได้รับ

[r:EXT\_SIO\_RCV] (อ่านอย่างเดียว)

◆ ฟังก์ชันรับ

IO\_READ ([p:EXT\_SIO], ตำแหน่งจัดเก็บของ LS, จำนวนไบต์)

◆ ฟังก์ชันส่ง

IO\_WRITE ([p:EXT\_SIO], ตำแหน่งจัดเก็บของ LS, จำนวนไบต์)

## ■ ตัวแปรควบคุม

| รายการ     | คำอธิบาย                                                                                                                               |
|------------|----------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ตัวแปรควบคุมนี้ใช้สำหรับล้างข้อมูลในบัฟเฟอร์การส่งข้อมูล บัฟเฟอร์การรับข้อมูล และสถานะข้อผิดพลาด ตัวแปรควบคุมนี้เป็นแบบเขียนอย่างเดียว |
| รูปแบบ     | เมื่อกำหนดบิต: [c:EXT_SIO_CTRL**] (**: 00 ถึง 15)<br>เมื่อกำหนดเวิร์ด: [c:EXT_SIO_CTRL]                                                |

### ตัวอย่างนิพจน์

เมื่อกำหนดบิต: [c:EXT\_SIO\_CTRL00] = 1

เมื่อกำหนดเวิร์ด: [c:EXT\_SIO\_CTRL] = 0x0007

### EXT\_SIO\_CTRL

|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| บิต | รายละเอียด                   |
|-----|------------------------------|
| 15  | สำรอง                        |
| 14  |                              |
| 13  |                              |
| 12  |                              |
| 11  |                              |
| 10  |                              |
| 9   |                              |
| 8   |                              |
| 7   |                              |
| 6   |                              |
| 5   |                              |
| 4   |                              |
| 3   | 1: ล้างการรับที่เก็ทโทมโฮสต์ |
| 2   | 1: ล้างข้อผิดพลาด            |
| 1   | 1: ล้างบัฟเฟอร์การรับข้อมูล  |
| 0   | 1: ล้างบัฟเฟอร์การส่งข้อมูล  |

- หมายเหตุ**
- เมื่อกำหนดเวิร์ด (เมื่อตั้งค่าบิตพร้อมกันสองบิตขึ้นไป) จะทำการประมวลผลตามลำดับดังนี้  
ล้างข้อผิดพลาด → ล้างข้อมูลในบัฟเฟอร์การรับข้อมูล → ล้างข้อมูลในบัฟเฟอร์การส่งข้อมูล

## ■ สถานะ

| รายการ     | คำอธิบาย                                                                                |
|------------|-----------------------------------------------------------------------------------------|
| ข้อมูลสรุป | สถานะจะรวมถึงข้อมูลต่อไปนี้<br>ตัวแปรสถานะนี้เป็นแบบเขียนอย่างเดียว                     |
| รูปแบบ     | เมื่อกำหนดบิต: [s:EXT_SIO_STAT**] (**: 00 ถึง 15)<br>เมื่อกำหนดเวิร์ด: [s:EXT_SIO_STAT] |

### ตัวอย่างนิพจน์

เมื่อกำหนดบิต: if ([s:EXT\_SIO\_STAT00] == 1)

เมื่อกำหนดเวิร์ด: if (([s:EXT\_SIO\_STAT] & 0x0001) <> 0)

### รายละเอียดของ EXT\_SIO\_STAT

|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| บิต | รายละเอียด                                                                        |
|-----|-----------------------------------------------------------------------------------|
| 15  | 0: ไม่มี D-Script/Global D-Script<br>1: มี D-Script/Global D-Script               |
| 14  | 0: ไม่มี extended script<br>1: มี Extended script                                 |
| 13  | สำรอง                                                                             |
| 12  |                                                                                   |
| 11  |                                                                                   |
| 10  |                                                                                   |
| 9   |                                                                                   |
| 8   |                                                                                   |
| 7   |                                                                                   |
| 6   |                                                                                   |
| 5   |                                                                                   |
| 4   | 0: ปกติ<br>1: การรับเกิดใหม่เอดส์                                                 |
| 3   | 0: ปกติ<br>1: การรับเกิดข้อผิดพลาด                                                |
| 2   | 0: ไม่มีข้อมูลการรับ<br>1: มีข้อมูลการรับ                                         |
| 1   | 0: ปกติ<br>1: การเกิดข้อผิดพลาด                                                   |
| 0   | 0: มีข้อมูลอยู่ในบัฟเฟอร์การส่งข้อมูล<br>1: ไม่มีข้อมูลอยู่ในบัฟเฟอร์การส่งข้อมูล |

#### หมายเหตุ

- บิตที่สำรองไว้อาจถูกกำหนดในภายหลัง ดังนั้น ให้กำหนดเฉพาะบิตที่จำเป็นเท่านั้น
- ข้อผิดพลาดในการส่งมีอยู่สองชนิด ได้แก่ ข้อผิดพลาดเนื่องจากการส่งเกิดใหม่เอดส์ และข้อผิดพลาดเนื่องจากการรับข้อมูลการส่งเต็ม เมื่อเกิดข้อผิดพลาดใดขึ้น บิตข้อผิดพลาดในการส่งจะเปิดขึ้นระยะเวลาใหม่เอดส์ของการส่งคือห้าวินาที
- ข้อผิดพลาดในการรับมีอยู่สี่ชนิด ได้แก่ ข้อผิดพลาดพาร์ตี, ข้อผิดพลาดโอเวอร์รัน, ข้อผิดพลาดเฟรมมิ่ง และโอเวอร์โฟลว์ เมื่อเกิดข้อผิดพลาดใดขึ้น บิตข้อผิดพลาดในการรับจะเปิดขึ้น
- หากตรวจพบข้อผิดพลาดในการส่ง ข้อมูลการส่งจะยังคงอยู่ในบัฟเฟอร์การส่งข้อมูล หากตรวจไม่พบข้อผิดพลาดในการส่ง ข้อมูลการส่งจะถูกส่งออกจากบัฟเฟอร์การส่งข้อมูล
- เมื่อใช้พอร์ตอินเทอร์เฟซแบบอนุกรม COM2 ซึ่งเป็น RS-422 จะตรวจไม่พบสัญญาณ CS (CTS) ดังนั้น จึงไม่สามารถตรวจพบได้ว่ามีสายเคเบิลหลุด

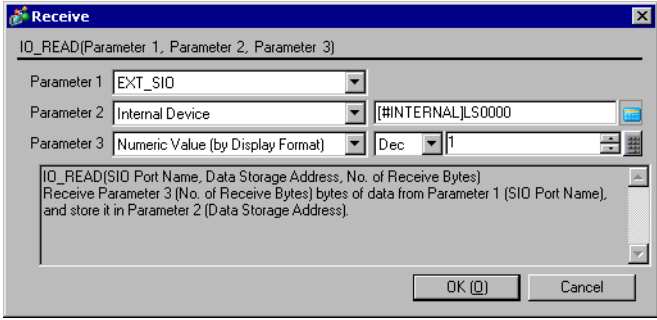
## ■ จำนวนข้อมูลที่ได้รับ

| รายการ     | คำอธิบาย                                                                                                 |
|------------|----------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | แสดงปริมาณข้อมูล (จำนวนไบต์) ที่ได้รับในคราวนั้น ๆ ขนาดข้อมูลที่ได้รับจะมีคุณสมบัติเป็นแบบอ่านอย่างเดียว |
| รูปแบบ     | [r:EXT_SIO_RECV]                                                                                         |

### ข้อสำคัญ

- ชื่อป้ายชื่อของจำนวนข้อมูลที่ได้รับ (จำนวนไบต์)  
เมื่อใช้ GP-PRO/PB III V. 6.0 และเวอร์ชันก่อนหน้า ชื่อป้ายชื่อซึ่งกำหนดไว้สำหรับขนาดข้อมูลที่ได้รับ คือ [r: EXT\_SIO\_RCV] อย่างไรก็ตาม คุณไม่จำเป็นต้องแก้ไขคำอธิบายเพราะจะมีฟังก์ชันเหมือนกัน ไม่ว่าจะเลือก [r: EXT\_SIO\_RCV] หรือ [r: EXT\_SIO\_RECV]

## 21.4.2 Receive

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                         |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | เขียนข้อความคำสั่งต่อไปนี้ขณะอ่านข้อมูลที่ได้รับจาก Extended SIO                                                                                                                                                                                                 |
| รูปแบบ     | <p>IO_READ ([p:EXT_SIO], ตำแหน่งจัดเก็บข้อมูล, จำนวนไบต์ที่ได้รับ)</p>  <p>Parameter 1: EXT_SIO<br/>         Parameter 2: อุปกรณ์ภายใน<br/>         Parameter 3: ค่าตัวเลข</p> |

### ตัวอย่างนิพจน์

IO\_READ ([p:EXT\_SIO], [w:[#INTERNAL]LS0100], 10)

จากตัวอย่างข้างบน จำนวนไบต์ที่ได้รับจะจัดเก็บไว้ใน LS0100 โดยจัดเก็บข้อมูลจำนวน 10 ไบต์โดยเริ่มตั้งแต่ LS0101T ภาพด้านล่างแสดงข้อมูลที่ได้รับซึ่งจัดเก็บไว้

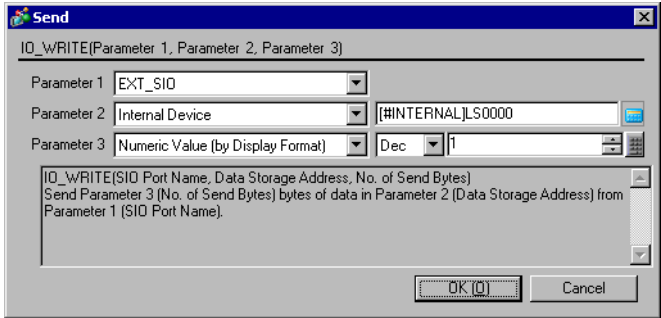
#### หมายเหตุ

- ในระหว่างการรับข้อมูลสามารถถ่ายโอนไบต์ได้เป็นจำนวนสูงสุด 2,011 ไบต์ ระบบจะเขียนข้อมูลหน่วยละ 1 ไบต์ลงในตำแหน่งเวิร์ดแต่ละตำแหน่ง

| ขนาดข้อมูลที่ได้รับ | ... | 10 ไบต์ |
|---------------------|-----|---------|
| LS0100              | 00  | ไบต์ 1  |
| LS0101              | 00  | ไบต์ 2  |
| LS0102              | 00  | ไบต์ 3  |
| LS0103              | 00  | ไบต์ 4  |
| LS0104              | 00  | ไบต์ 5  |
| LS0105              | 00  | ไบต์ 6  |
| LS0106              | 00  | ไบต์ 7  |
| LS0107              | 00  | ไบต์ 8  |
| LS0108              | 00  | ไบต์ 9  |
| LS0109              | 00  | ไบต์ 10 |

วิธีการจัดเก็บข้อมูลที่ได้รับ

### 21.4.3 Send

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                       |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | เขียนข้อความคำสั่งต่อไปนี้เมื่อเขียนข้อมูลลงใน Extended SIO                                                                                                                                                                                    |
| รูปแบบ     | <p>IO_WRITE ([p:EXT_SIO], ตำแหน่งจัดเก็บข้อมูล, จำนวนไบต์ที่ส่ง)</p>  <p>Parameter 1: EXT_SIO<br/> Parameter 2: อุปกรณ์ภายใน<br/> Parameter 3: ค่าตัวเลข</p> |

#### ตัวอย่างนิพจน์

IO\_WRITE ([p:EXT\_SIO], [w:[#INTERNAL]LS0100], 10)

จากตัวอย่างข้างบน จะเป็นการส่งข้อมูล 10 ไบต์โดยเริ่มจาก LS0100 ภาพด้านล่างแสดงข้อมูลที่ส่งออกไปซึ่งจัดเก็บไว้

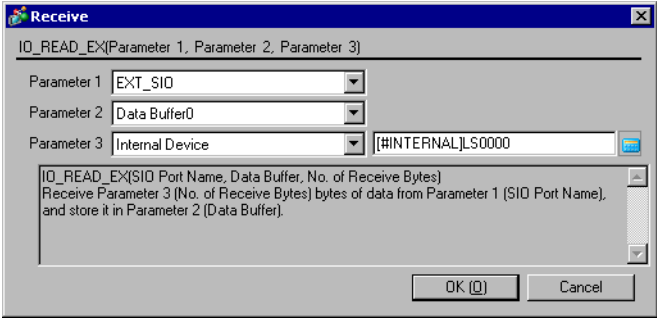
#### หมายเหตุ

- ในระหว่างการรับข้อมูลสามารถถ่ายโอนไบต์ได้เป็นจำนวนสูงสุด 2,012 ไบต์
- เมื่อใช้อุปกรณ์ LS สำหรับบัฟเฟอร์การส่งข้อมูล ระบบจะเขียนข้อมูลเป็นไบต์เดียวลงในตำแหน่งเวิร์ดแต่ละตำแหน่ง

|        |    |         |
|--------|----|---------|
| LS0100 | 00 | ไบต์ 1  |
| LS0101 | 00 | ไบต์ 2  |
| LS0102 | 00 | ไบต์ 3  |
| LS0103 | 00 | ไบต์ 4  |
| LS0104 | 00 | ไบต์ 5  |
| LS0105 | 00 | ไบต์ 6  |
| LS0106 | 00 | ไบต์ 7  |
| LS0107 | 00 | ไบต์ 8  |
| LS0108 | 00 | ไบต์ 9  |
| LS0109 | 00 | ไบต์ 10 |

วิธีการจัดเก็บข้อมูลที่ส่ง

#### 21.4.4 Extended Receive

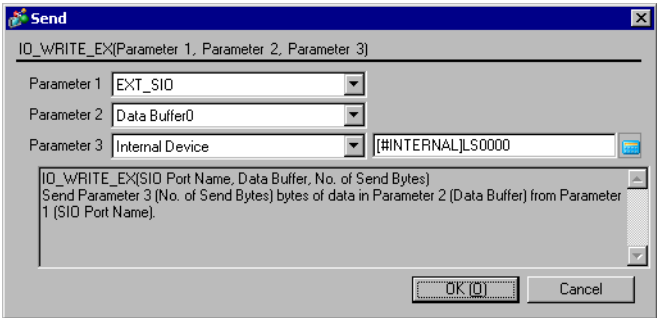
| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                    |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | รับข้อมูลจาก Extended SIO ตามขนาดที่ระบุในขนาดข้อมูลที่ได้รับ (ไบต์) แล้วเก็บไว้ในบัฟเฟอร์ข้อมูล รับข้อมูลจาก Extended SIO ตามจำนวนไบต์ที่ระบุใน Parameter 3 และจัดเก็บไว้ในบัฟเฟอร์ข้อมูลที่ระบุใน Parameter 2<br>ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น                                                                                          |
| รูปแบบ     | <p>IO_READ_EX ([p:EXT_SIO], บัฟเฟอร์ข้อมูล, จำนวนไบต์ที่ได้รับ)</p>  <p>Parameter 1: [p:EXT_SIO]<br/>         Parameter 2: บัฟเฟอร์ข้อมูล<br/>         Parameter 3: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 3 คือตั้งแต่ 1 ถึง 1,024)</p> |

##### ตัวอย่างนิพจน์

IO\_READ\_EX ([p:EXT\_SIO], databuf1, 10)

จากตัวอย่างข้างบน เป็นการรับข้อมูลจำนวน 10 ไบต์จาก Extended SIO แล้วจัดเก็บไว้ใน “databuf1”

## 21.4.5 Extended Send

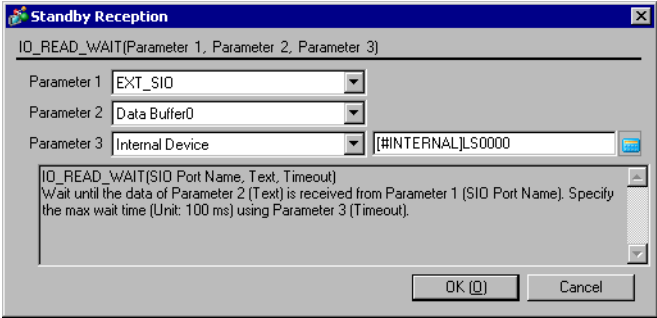
| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                  |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ส่งข้อมูลในบัฟเฟอร์ข้อมูลด้วย Extended SIO ตามขนาดของจำนวนไบต์ที่ส่ง Extended SIO จะส่งรายละเอียดของบัฟเฟอร์ข้อมูลที่ระบุใน Parameter 2 โดยมีความยาวตามที่ระบุใน Parameter 3 ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น                                                                                                                              |
| รูปแบบ     | <p>IO_WRITE_EX ([p:EXT_SIO], บัฟเฟอร์ข้อมูล, จำนวนไบต์ที่ส่ง)</p>  <p>Parameter 1: [p:EXT_SIO]<br/>         Parameter 2: บัฟเฟอร์ข้อมูล<br/>         Parameter 3: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 3 คือตั้งแต่ 1 ถึง 1,024)</p> |

### ตัวอย่างนิพจน์

IO\_WRITE\_EX ([p:EXT\_SIO], databuf0, 10)

จากตัวอย่างข้างบน เป็นการส่งข้อมูลใน “databuf0” 10 ไบต์จาก Extended SIO

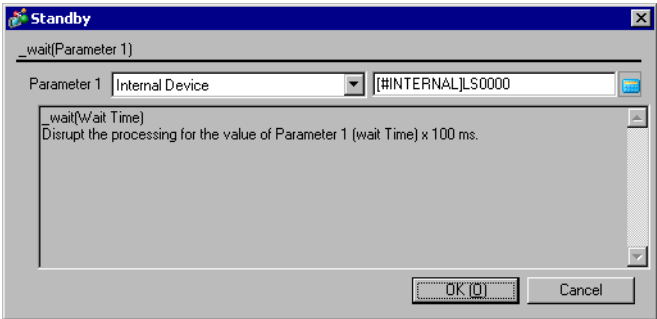
## 21.4.6 ฟังก์ชัน Standby Reception

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | <p>ระบบอยู่ในโหมดสแตนด์บายการรับจนกว่าจะได้รับข้อความที่ระบุ เมื่อระยะเวลาไทม์เอาต์สิ้นสุดลง ระบบจะตั้งค่าให้บิต 4 (ข้อผิดพลาดเนื่องจากการรับเกิดไทม์เอาต์) ของสถานะ [s: EXT_SIO_STAT] คุณสามารถตั้งระยะเวลาไทม์เอาต์ได้โดยเพิ่มขึ้นครั้งละ 100 มิลลิวินาที ระบบจะอยู่ในโหมดสแตนด์บายการรับจนกว่าจะได้รับสตริงอักขระหรือรหัสอักขระที่ระบุใน Parameter 2 ระยะเวลาไทม์เอาต์จะกำหนดด้วย Parameter 3</p> <p>ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น</p> |
| รูปแบบ     | <p>IO_READ_WAIT([p:EXT_SIO], ข้อความ, ไทม์เอาต์)</p>  <p>Parameter 1: [p:EXT_SIO]<br/>         Parameter 2: ค่าตัวเลข, ข้อความ, บัฟเฟอร์ข้อมูล<br/>         Parameter 3: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 3 คือตั้งแต่ 1 ถึง 600)</p>                                                                                              |

### ข้อสำคัญ

- ข้อมูลที่รับไม่สามารถใช้ได้จนกว่าจะได้รับข้อความที่ระบุ (มิฉะนั้น ระบบจะยกเลิกข้อมูลดังกล่าว)
- สามารถระบุอักขระได้สูงสุด 128 ตัว (ไบต์) โปรดทราบว่าหากระบุสตริงมากกว่าที่กำหนดไว้ จะไม่สามารถสแตนด์บายการรับได้

## 21.4.7 ฟังก์ชัน Standby

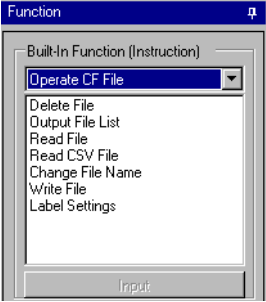
| รายการ     | คำอธิบาย                                                                                                                                                                                                                     |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ระบบหยุดรอ (พักการทำงานชั่วคราว) เป็นระยะเวลาตามที่ระบุไว้ คุณสามารถตั้งเวลาได้โดยเพิ่มขึ้นครั้งละ 100 มิลลิวินาที<br>ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น                                                        |
| รูปแบบ     | <p>_wait (เวลารอ)</p>  <p>Parameter 1: อุปกรณ์ภายใน, ตำแหน่งชั่วคราว, ค่าตัวเลข (ช่วงที่ใช้ได้สำหรับ Parameter 1 คือตั้งแต่ 1 ถึง 600)</p> |

### ตัวอย่างนิพจน์

\_wait (10)

จากตัวอย่างข้างบน ระบบจะรอเป็นเวลาหนึ่งวินาที

## 21.5 การใช้งานไฟล์ในการ์ด CF

| Operate CF File                                                                   | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Label Settings<br> “21.5.1 Label Settings” (หน้า 21-39)<br>กำหนดได้จากจำนวนไฟล์ที่แสดง, จำนวนไบต์ที่อ่าน และสถานะข้อผิดพลาดของการ์ด CF                 |
|                                                                                   | Write File<br> “21.5.2 Write File” (หน้า 21-46)<br>เขียนข้อมูลตามจำนวนไบต์ที่ระบุจากตำแหน่งต้นทางไปยังไฟล์ที่ระบุ                                      |
|                                                                                   | Change File Name<br> “21.5.3 Change File Name” (หน้า 21-50)<br>แก้ไขชื่อไฟล์                                                                           |
|                                                                                   | Read CSV File<br> “21.5.4 Read CSV File” (หน้า 21-52)<br>อ่านข้อมูลในเซลล์จากไฟล์ CSV และเขียนลงในตำแหน่งเวิร์ด                                        |
|                                                                                   | Read File<br> “21.5.5 Read File” (หน้า 21-55)<br>อ่านข้อมูลที่อยู่ต่อจากออฟเซตที่ระบุไว้ตามจำนวนไบต์ที่ระบุ และเขียนข้อมูลที่อ่านได้ลงในตำแหน่งปลายทาง |
|                                                                                   | Output File List<br> “21.5.6 Output File List” (หน้า 21-57)<br>เขียนรายการไฟล์ที่มีอยู่ในโฟลเดอร์ที่ระบุลงในอุปกรณ์ภายใน                               |
|                                                                                   | Delete File<br> “21.5.7 Delete File” (หน้า 21-58)<br>ลบไฟล์ทิ้ง                                                                                      |

## 21.5.1 Label Settings

สถานะของการ์ด CF จะมีสถานะต่างๆ ต่อไปนี้

| ชื่อสถานะ                  | ชื่อป้ายชื่อ        | คำอธิบาย                                                                            |
|----------------------------|---------------------|-------------------------------------------------------------------------------------|
| จำนวนไฟล์ที่แสดง           | [s:CF_FILELIST_NUM] | จัดเก็บจำนวนไฟล์ตามที่แสดงจริงเมื่อมีการเรียกใช้ฟังก์ชันส่งรายการไฟล์ “_CF_dir ( )” |
| จำนวนไบต์ที่อ่าน           | [s:CF_READ_NUM]     | จัดเก็บจำนวนไบต์ที่สามารถอ่านได้เมื่อมีการเรียกใช้ฟังก์ชันอ่านไฟล์ “_CF_read ( )”   |
| สถานะข้อผิดพลาดของการ์ด CF | [s:CF_ERR_STAT]     | จัดเก็บสถานะข้อผิดพลาดที่สร้างขึ้นเมื่อมีการเข้าใช้ข้อมูลในการ์ด CF                 |

### ■ จำนวนไฟล์ที่แสดง

เมื่อเรียกใช้ฟังก์ชันส่งรายการไฟล์ “\_CF\_dir ( )” จำนวนรายการไฟล์ที่ถูกเขียนจริงในพื้นที่ LS จะถูกจัดเก็บไว้ใน “จำนวนไฟล์ที่แสดง [s:CF\_FILELIST\_NUM]”

#### ตัวอย่างการใช้

```
_CF_dir (“\DATA\*.*”, [w:[#INTERNAL]LS0100], 10, 0)
[w:LS0200] = [s:CF_FILELIST_NUM]
```

```
\DATA — DATA0000.BIN
        — DATA0001.BIN
        — DATA0002.BIN
        — DATA0003.BIN
        — DATA0004.BIN
```

เมื่อมีความพยายามขอรับรายการไฟล์ที่ประกอบด้วยไฟล์จำนวน 10 ไฟล์ แต่โฟลเดอร์ที่ระบุไว้มีไฟล์อยู่เพียง 5 ไฟล์ ระบบจะจัดเก็บค่า “5” ไว้ใน [s:CF\_FILELIST\_NUM]

#### ข้อสำคัญ

- เมื่อไม่มีไฟล์สำหรับเขียน ระบบจะเขียนจำนวนไฟล์ทั้งหมดที่มีอยู่ในโฟลเดอร์ที่ระบุลงใน [s:CF\_FILELIST\_NUM]

### ■ จำนวนไบต์ที่อ่าน

เมื่อเรียกใช้ฟังก์ชันอ่านไฟล์ “\_CF\_read ( )” จำนวนไบต์ที่อ่านจริงจะถูกจัดเก็บไว้ใน “จำนวนไบต์ที่อ่าน [s:CF\_READ\_NUM]”

#### ตัวอย่างการใช้

```
_CF_read (“\DATA”, “\DATA0001.BIN”, [w:[#INTERNAL]LS0100], 16, 16)
[w:[#INTERNAL]LS0200] = [s:CF_READ_NUM]
```

เมื่อมีความพยายามขออ่านข้อมูลจำนวน 16 ไบต์ แต่อ่านได้สำเร็จเพียง 12 ไบต์ ระบบจะจัดเก็บค่า “12” ไว้ใน [s:CF\_READ\_NUM]

## ■ สถานะข้อผิดพลาดของการ์ด CF

จัดเก็บสถานะข้อผิดพลาดที่สร้างขึ้นเมื่อมีการเข้าใช้ข้อมูลในการ์ด CF

| ตำแหน่งบิต | ชื่อข้อผิดพลาด                 | คำอธิบาย                                                                                                                                                                                                                                                |
|------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15         | สำรอง                          | สำรอง                                                                                                                                                                                                                                                   |
| 14         |                                |                                                                                                                                                                                                                                                         |
| 13         |                                |                                                                                                                                                                                                                                                         |
| 12         |                                |                                                                                                                                                                                                                                                         |
| 11         |                                |                                                                                                                                                                                                                                                         |
| 10         |                                |                                                                                                                                                                                                                                                         |
| 9          |                                |                                                                                                                                                                                                                                                         |
| 8          |                                |                                                                                                                                                                                                                                                         |
| 7          |                                |                                                                                                                                                                                                                                                         |
| 6          | ข้อผิดพลาดในการเปลี่ยนชื่อไฟล์ | <ul style="list-style-type: none"> <li>• การ์ด CF ถูกดึงออกขณะทำงานอยู่</li> <li>• ไม่มีไฟล์ที่ระบุ</li> <li>• มีความพยายามที่จะเปลี่ยนชื่อไฟล์ที่เป็นไฟล์แบบอ่านอย่างเดียว</li> </ul>                                                                  |
| 5          | ข้อผิดพลาดในการลบไฟล์          | <ul style="list-style-type: none"> <li>• การ์ด CF ถูกดึงออกขณะทำงานอยู่</li> <li>• ไม่มีไฟล์ที่ระบุ</li> <li>• มีความพยายามที่จะลบไฟล์ที่เป็นไฟล์แบบอ่านอย่างเดียว</li> </ul>                                                                           |
| 4          | ข้อผิดพลาดในการเขียนไฟล์       | <ul style="list-style-type: none"> <li>• การ์ด CF ถูกดึงออกขณะทำงานอยู่</li> <li>• ไม่มีพื้นที่ว่างเหลือในการ์ด CF</li> <li>• มีความพยายามที่จะเขียนข้อมูลลงในไฟล์ที่เป็นไฟล์แบบอ่านอย่างเดียว</li> <li>• ไม่มีไฟล์ตามที่กำหนดให้ “เขียนทับ”</li> </ul> |
| 3          | ข้อผิดพลาดในการอ่านไฟล์        | <ul style="list-style-type: none"> <li>• การ์ด CF ถูกดึงออกขณะทำงานอยู่</li> <li>• ไม่มีไฟล์ที่ระบุ</li> </ul>                                                                                                                                          |
| 2          | ข้อผิดพลาดในรายการไฟล์         | <ul style="list-style-type: none"> <li>• การ์ด CF ถูกดึงออกขณะทำงานอยู่</li> <li>• ไม่มีไฟล์เดอ์ที่ระบุ</li> </ul>                                                                                                                                      |
| 1          | ข้อผิดพลาดของการ์ด CF          | <ul style="list-style-type: none"> <li>• การ์ด CF ใช้ไม่ได้</li> <li>• สื่อบันทึกข้อมูลที่เสียบไว้ไม่ใช่การ์ด CF</li> </ul>                                                                                                                             |
| 0          | ไม่มีการ์ด CF                  | <ul style="list-style-type: none"> <li>• ไม่ได้เสียบการ์ด CF</li> <li>• ฝาครอบเปิดอยู่</li> </ul>                                                                                                                                                       |

- แม้จะเกิดข้อผิดพลาดในการ์ด CF แต่การทำงานยังคงดำเนินต่อไป ทุกครั้งที่คุณใช้ฟังก์ชันการใช้งานไฟล์ในการ์ด CF ต้องเขียนสคริปต์สำหรับตรวจสอบข้อผิดพลาดด้วยตัวอย่าง

```

_CF_dir ("\\DATA\\*.*", [w:[#INTERNAL]LS0100], 2, 1) // ส่งข้อมูลรายการไฟล์
if([s:CF_ERR_STAT02] <> 0) // ตรวจสอบสถานะข้อผิดพลาด
{
    set ([b:[#INTERNAL]LS005000]) // กำหนดตำแหน่งบิตสำหรับแสดงข้อผิดพลาด
}
endif

```

### รายละเอียดสถานะข้อผิดพลาดของการ์ด CF - พื้นที่จัดเก็บข้อมูล

หากเกิดข้อผิดพลาดขึ้น ระบบจะตั้งค่าบิตตามที่เหมาะสม คุณสามารถตรวจสอบสาเหตุของข้อผิดพลาดนั้นได้ โดยที่รายละเอียดสถานะ รายละเอียดสถานะสำหรับแต่ละฟังก์ชันจะจัดเก็บไว้ในพื้นที่ LS9132 จนถึง LS9137 ของพื้นที่เสริมของระบบ พื้นที่เหล่านี้เป็นแบบอ่านอย่างเดียว

| พื้นที่ LS |                               |
|------------|-------------------------------|
| LS0000     |                               |
| :          |                               |
| LS9132     | สถานะรายการไฟล์ในการ์ด CF     |
| LS9133     | สถานะการอ่านข้อมูลในการ์ด CF  |
| LS9134     | สถานะการเขียนข้อมูลในการ์ด CF |
| LS9135     | สถานะการลบข้อมูลในการ์ด CF    |
| LS9136     | สถานะการเปลี่ยนชื่อในการ์ด CF |
| LS9137     | สถานะการอ่านข้อมูล CSV        |
| :          |                               |
| LS9999     |                               |

### รายการข้อผิดพลาดสำหรับแต่ละฟังก์ชัน

| ชื่อฟังก์ชันตัวแก้ไข |        | สถานะข้อผิดพลาด | สาเหตุ                                                                                             |
|----------------------|--------|-----------------|----------------------------------------------------------------------------------------------------|
| _CF_dir ( )          | LS9132 | 0010h           | ข้อมูล D-Script ไม่ถูกต้อง (เกิดข้อผิดพลาดในการค้นชื่อไฟล์เตอร์ที่ระบุด้วยสตริงแบบตายตัว)          |
|                      |        | 0012h           | ชื่อไฟล์ (ชื่อพาธ) มีข้อผิดพลาด                                                                    |
|                      |        | 0018h           | ช่วงการเขียนของพื้นที่ LS เกิดข้อผิดพลาด                                                           |
|                      |        | 0020h           | ไม่มีการ์ด CF                                                                                      |
|                      |        | 0021h           | การ์ด CF ใช้ไม่ได้                                                                                 |
|                      |        | 0100h           | การเปิดไดเรกทอรีเกิดข้อผิดพลาด                                                                     |
| _CF_read ( )         | LS9133 | 0010h           | ข้อมูล D-Script ไม่ถูกต้อง (เกิดข้อผิดพลาดในการค้นชื่อไฟล์เตอร์/ชื่อไฟล์ที่ระบุด้วยสตริงแบบตายตัว) |
|                      |        | 0011h           | ช่วงการอ่านของพื้นที่ LS เกิดข้อผิดพลาด                                                            |
|                      |        | 0012h           | ชื่อไฟล์ (ชื่อพาธ) มีข้อผิดพลาด                                                                    |
|                      |        | 0018h           | ช่วงการเขียนของพื้นที่ LS เกิดข้อผิดพลาด                                                           |
|                      |        | 0020h           | ไม่มีการ์ด CF                                                                                      |
|                      |        | 0021h           | การ์ด CF ใช้ไม่ได้                                                                                 |
|                      |        | 0101h           | การหาไฟล์เกิดข้อผิดพลาด (ข้อผิดพลาดของออฟเซต)                                                      |
|                      |        | 0102h           | จำนวนไบต์ที่อ่านมีข้อผิดพลาด                                                                       |
|                      |        | 0110h           | การสร้าง (เปิด) ไฟล์เกิดข้อผิดพลาด                                                                 |

ต่อ

| ชื่อฟังก์ชันตัวแก้ไข |        | สถานะข้อผิดพลาด | สาเหตุ                                                                                                      |
|----------------------|--------|-----------------|-------------------------------------------------------------------------------------------------------------|
| _CF_write ( )        | LS9134 | 0010h           | ข้อมูล D-Script ไม่ถูกต้อง (เกิดข้อผิดพลาดในการค้นชื่อไฟล์เตอร์/ชื่อไฟล์ที่ระบุด้วยสตริงแบบตายตัว)          |
|                      |        | 0011h           | ช่วงการอ่านของพื้นที่ LS เกิดข้อผิดพลาด                                                                     |
|                      |        | 0012h           | ชื่อไฟล์ (ชื่อพาธ) มีข้อผิดพลาด                                                                             |
|                      |        | 0020h           | ไม่มีการ์ด CF                                                                                               |
|                      |        | 0021h           | การ์ด CF ใช้ไม่ได้                                                                                          |
|                      |        | 0101h           | การหาไฟล์เกิดข้อผิดพลาด (ข้อผิดพลาดของออฟเซต)                                                               |
|                      |        | 010h            | การสร้างไฟล์เกิดข้อผิดพลาด                                                                                  |
|                      |        | 0108h           | โหมดการเขียนเกิดข้อผิดพลาด                                                                                  |
|                      |        | 0110h           | การสร้าง (เปิด) ไฟล์เกิดข้อผิดพลาด                                                                          |
|                      |        | 0111h           | การเขียนไฟล์เกิดข้อผิดพลาด (เช่น การ์ด CF มีพื้นที่ว่างไม่เพียงพอ)                                          |
| _CF_delete ( )       | LS9135 | 0010h           | ข้อมูล D-Script ไม่ถูกต้อง (เกิดข้อผิดพลาดในการค้นชื่อไฟล์เตอร์/ชื่อไฟล์ที่ระบุด้วยสตริงแบบตายตัว)          |
|                      |        | 0011h           | ช่วงการอ่านของพื้นที่ LS เกิดข้อผิดพลาด                                                                     |
|                      |        | 0012h           | ชื่อไฟล์ (ชื่อพาธ) มีข้อผิดพลาด                                                                             |
|                      |        | 0020h           | ไม่มีการ์ด CF                                                                                               |
|                      |        | 0021h           | การ์ด CF ใช้ไม่ได้                                                                                          |
|                      |        | 0112h           | การลบไฟล์เกิดข้อผิดพลาด (เช่น ไม่มีไฟล์ที่ระบุไฟล์ที่ระบุเป็นแบบอ่านอย่างเดียว)                             |
| _CF_rename ( )       | LS9136 | 0010h           | ข้อมูล D-Script ไม่ถูกต้อง (เกิดข้อผิดพลาดในการค้นชื่อไฟล์เตอร์/ชื่อไฟล์ที่ระบุด้วยสตริงแบบตายตัว)          |
|                      |        | 0011h           | ช่วงการอ่านของพื้นที่ LS เกิดข้อผิดพลาด                                                                     |
|                      |        | 0012h           | ชื่อไฟล์ (ชื่อพาธ) มีข้อผิดพลาด                                                                             |
|                      |        | 0020h           | ไม่มีการ์ด CF                                                                                               |
|                      |        | 0021h           | การ์ด CF ใช้ไม่ได้                                                                                          |
|                      |        | 0114h           | การเปลี่ยนชื่อไฟล์เกิดข้อผิดพลาด (เช่น ไม่มีไฟล์ที่ระบุไฟล์ที่ระบุเป็นแบบอ่านอย่างเดียว มีชื่อไฟล์นั้นแล้ว) |
| _CFread_csv()        | LS9137 | 0001h           | พารามิเตอร์เกิดข้อผิดพลาด                                                                                   |
|                      |        | 000h            | ข้อผิดพลาดของการ์ด CF (ไม่มีการ์ด CF, การเปิดไฟล์เกิดข้อผิดพลาด, การอ่านไฟล์เกิดข้อผิดพลาด)                 |
|                      |        | 0003h           | การเขียนเกิดข้อผิดพลาด                                                                                      |

## โหมดการจัดเก็บข้อมูล

เมื่อมีการอ่าน/เขียนข้อมูลจาก/ไปยังตำแหน่งอุปกรณ์ ขณะเรียกใช้ฟังก์ชันอ่านไฟล์/เขียนไฟล์ คุณสามารถระบุลำดับการจัดเก็บของข้อมูลที่เขียน (อ่าน) ได้ การตั้งค่าโหมดการจัดเก็บข้อมูลในพื้นที่ LS9130 สามารถเปลี่ยนลำดับการจัดเก็บได้ โดยสามารถเลือกโหมดได้จากตัวเลือกทั้งสี่ ได้แก่ โหมด 0, 1, 2 และ 3

### ◆ โหมด 0

ตัวอย่าง การใช้ฟังก์ชันอ่านไฟล์เขียนสตริง “ABCDEFGH” ลงในตำแหน่งอุปกรณ์

[w:[#INTERNAL]LS9130] = 0

\_CF\_read (“DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 0, 7)

- เมื่อตำแหน่งอุปกรณ์ยาว 16 บิต

|        |     |     |  |  |
|--------|-----|-----|--|--|
| LS0100 | 'A' | 'B' |  |  |
| LS0101 | 'C' | 'D' |  |  |
| LS0102 | 'E' | 'F' |  |  |
| LS0103 | 'G' | 0   |  |  |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

- เมื่อตำแหน่งอุปกรณ์ยาว 32 บิต

|        |     |     |     |     |
|--------|-----|-----|-----|-----|
| LS0100 | 'A' | 'B' | 'C' | 'D' |
| LS0101 | 'E' | 'F' | 'G' | 0   |
| LS0102 |     |     |     |     |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

### ◆ โหมด 1

ตัวอย่าง การใช้ฟังก์ชันอ่านไฟล์เขียนสตริง “ABCDEFGH” ลงในตำแหน่งอุปกรณ์

[w:[#INTERNAL]LS9130] = 1

\_CF\_read (“DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 0, 7)

- เมื่อตำแหน่งอุปกรณ์ยาว 16 บิต

|        |     |     |  |  |
|--------|-----|-----|--|--|
| LS0100 | 'B' | 'A' |  |  |
| LS0101 | 'D' | 'C' |  |  |
| LS0102 | 'F' | 'E' |  |  |
| LS0103 | 0   | 'G' |  |  |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

- เมื่อตำแหน่งอุปกรณ์ยาว 32 บิต

|        |     |     |     |     |
|--------|-----|-----|-----|-----|
| LS0100 | 'B' | 'A' | 'D' | 'C' |
| LS0101 | 'F' | 'E' | 0   | 'G' |
| LS0102 |     |     |     |     |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

## ◆ โหมด 2

ตัวอย่าง การใช้ฟังก์ชันอ่านไฟล์เขียนสตริง “ABCDEFGH” ลงในตำแหน่งอุปกรณ์

[w:[#INTERNAL]LS9130] = 2

\_CF\_read (“DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 0, 7)

- เมื่อตำแหน่งอุปกรณ์ยาว 16 บิต

|        |     |     |
|--------|-----|-----|
| LS0100 | 'C' | 'D' |
| LS0101 | 'A' | 'B' |
| LS0102 | 'G' | 0   |
| LS0103 | 'E' | 'F' |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

- เมื่อตำแหน่งอุปกรณ์ยาว 32 บิต

|        |     |     |     |     |
|--------|-----|-----|-----|-----|
| LS0100 | 'C' | 'D' | 'A' | 'B' |
| LS0101 | 0   | 'G' | 'E' | 'F' |
| LS0102 |     |     |     |     |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

## ◆ โหมด 3

ตัวอย่าง การใช้ฟังก์ชันอ่านไฟล์เขียนสตริง “ABCDEFGH” ลงในตำแหน่งอุปกรณ์

[w:[#INTERNAL]LS9130] = 3

\_CF\_read (“DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 0, 7)

- เมื่อตำแหน่งอุปกรณ์ยาว 16 บิต

|        |     |     |
|--------|-----|-----|
| LS0100 | 'D' | 'C' |
| LS0101 | 'B' | 'A' |
| LS0102 | 0   | 'G' |
| LS0103 | 'F' | 'E' |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

- เมื่อตำแหน่งอุปกรณ์ยาว 32 บิต

|        |     |     |     |     |
|--------|-----|-----|-----|-----|
| LS0100 | 'D' | 'C' | 'B' | 'A' |
| LS0101 | 0   | 'G' | 'F' | 'E' |
| LS0102 |     |     |     |     |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

### ข้อสำคัญ

- โหมดการจัดเก็บข้อมูลไม่เหมือนกับโหมดข้อมูลสตริงในการตั้งค่าระบบ ความสัมพันธ์ระหว่างโหมดการจัดเก็บข้อมูลกับโหมดข้อมูลสตริงจะแสดงอยู่ในตารางด้านล่าง

| ลำดับการจัดเก็บของอุปกรณ์ข้อมูล | ลำดับการจัดเก็บไบต์ในเวิร์ดแบบ LH หรือ HL | ลำดับการจัดเก็บในดับเบิลเวิร์ดแบบ LH หรือ HL | โหมดการจัดเก็บข้อมูล D-Script | โหมดข้อมูลตัวอักษร |
|---------------------------------|-------------------------------------------|----------------------------------------------|-------------------------------|--------------------|
| จัดเก็บจากข้อมูลแรกสุด          | ลำดับ HL                                  | ลำดับ HL                                     | 0                             | 1                  |
|                                 | ลำดับ LH                                  |                                              | 1                             | 2                  |
|                                 | ลำดับ HL                                  | ลำดับ LH                                     | 2                             | 5                  |
|                                 | ลำดับ LH                                  |                                              | 3                             | 4                  |
| จัดเก็บจากข้อมูลท้ายสุด         | ลำดับ HL                                  | ลำดับ HL                                     | —                             | 3                  |
|                                 | ลำดับ LH                                  |                                              | —                             | 7                  |
|                                 | ลำดับ HL                                  | ลำดับ LH                                     | —                             | 8                  |
|                                 | ลำดับ LH                                  |                                              | —                             | 6                  |

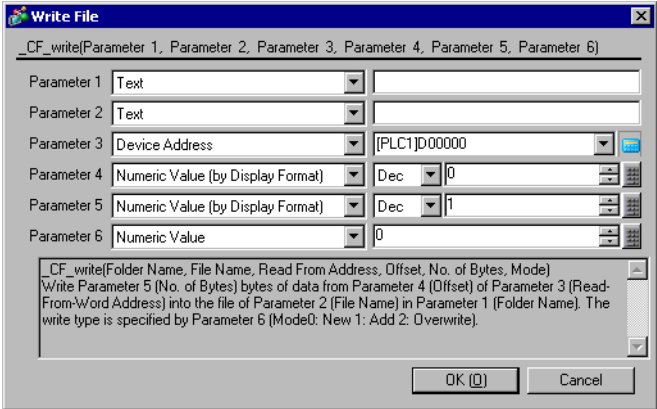
**ข้อสำคัญ**

- การเขียนข้อมูลในการ์ด CF มีข้อจำกัดเกี่ยวกับจำนวนครั้งที่สามารถเขียนซ้ำได้ โปรดสำรองข้อมูลทั้งหมดในการ์ด CF เก็บไว้ในสื่อเก็บข้อมูลอื่นอยู่เป็นประจำ (สามารถเขียนทับข้อมูลในรูปแบบ DOS ขนาด 500KB ได้ 100,000 ครั้ง)
- หากเกิดข้อผิดพลาดขึ้นระหว่างประมวลผลการ์ด CF ระบบจะเขียนข้อผิดพลาดลงในสถานะข้อผิดพลาดของการ์ด CF [s:CF\_ERR\_STAT] สำหรับรายละเอียดเพิ่มเติม โปรดดูที่ “■ สถานะข้อผิดพลาดของการ์ด CF” (หน้า 21-40)
- ชื่อไฟล์เดอร์หรือชื่อไฟล์ไม่สามารถใช้สัญลักษณ์และอักขระต่อไปนี้ หากใช้สัญลักษณ์และอักขระเหล่านี้ในชื่อไฟล์เดอร์หรือชื่อไฟล์จะทำให้เกิดข้อผิดพลาด

|   |   |   |   |               |   |   |
|---|---|---|---|---------------|---|---|
| : | , | = | + | /             | " | [ |
| ] |   | < | > | (พื้นที่ว่าง) | ? |   |

- หากต้องการระบุไฟล์เดอร์ราก (ไดเรกทอรี) ให้ระบุชื่อไฟล์เดอร์เป็น “ ” (สตริงเปล่า)

## 21.5.2 Write File

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | เขียนข้อมูลตามจำนวนไบต์ที่ระบุจากตำแหน่งต้นทางไปยังไฟล์ที่ระบุ โดยสามารถเลือกโหมดใดโหมดหนึ่งในสามโหมดนี้ได้แก่ “New”, “Add” หรือ “Overwrite” โปรดดูรายละเอียดเพิ่มเติมเกี่ยวกับลำดับการจัดเก็บข้อมูลได้ในส่วน “โหมดการจัดเก็บข้อมูล” ข้างล่าง                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| รูปแบบ     | <p>_CF_write (ชื่อโฟลเดอร์, ชื่อไฟล์, ตำแหน่งที่ถูกรอ่าน, ออฟเซต, จำนวนไบต์, โหมด)</p>  <p>Parameter 1<br/>ชื่อโฟลเดอร์: สตริงแบบตายตัว (ความยาวสูงสุด: อักขระแบบไบต์เดียว 32 อักขระ)</p> <p>Parameter 2<br/>ชื่อไฟล์: สตริงแบบตายตัว, อุปกรณ์ภายใน (ความยาวสูงสุด: อักขระแบบไบต์เดียว 32 อักขระ), อุปกรณ์ภายใน + ตำแหน่งชั่วคราว</p> <p>Parameter 3<br/>ตำแหน่งที่ถูกรอ่าน: ตำแหน่งอุปกรณ์, ตำแหน่งอุปกรณ์ + ตำแหน่งชั่วคราว</p> <p>Parameter 4<br/>ออฟเซต: ค่าตัวเลข, ตำแหน่งอุปกรณ์, ตำแหน่งชั่วคราว (จำนวนสูงสุดที่สามารถระบุได้: 65,535 สำหรับความยาว 16 บิต, 4,294,967,295 สำหรับความยาว 32 บิต)</p> <p>Parameter 5<br/>จำนวนไบต์: ค่าตัวเลข, ตำแหน่งอุปกรณ์, ตำแหน่งชั่วคราว (จำนวนสูงสุดที่สามารถระบุได้: 1,280)</p> <p>Parameter 6<br/>โหมด: ค่าตัวเลข, ตำแหน่งอุปกรณ์, ตำแหน่งชั่วคราว (ค่าที่ใช้ได้: 0, 1, 2)</p> |

## รูปแบบการจัดเก็บข้อมูล (โหมด)

| โหมด | ชื่อ      | คำอธิบาย                                                                                                                                                                                                                                                                                                                                  |
|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0    | New       | สร้างไฟล์ใหม่ หากมีไฟล์ที่มีชื่อเหมือนกันอยู่ ระบบจะลบไฟล์นั้นทิ้ง                                                                                                                                                                                                                                                                        |
| 1    | Add       | เพิ่มข้อมูลลงในไฟล์ที่ระบุ หากไม่มีไฟล์ที่ระบุ ระบบจะสร้างไฟล์ใหม่ขึ้นมา                                                                                                                                                                                                                                                                  |
| 2    | Overwrite | เขียนทับบางส่วนของไฟล์นั้น หากระบุออฟเซตใหญ่กว่าขนาดไฟล์ ระบบจะเติม 0 ลงในพื้นที่ที่เกินมาและเขียนข้อมูลต่อจากพื้นที่นั้น หากระบุออฟเซตไว้ตอนท้ายของข้อมูลไฟล์ จะเหมือนกับการเพิ่มข้อมูลลงในไฟล์ หากไม่มีไฟล์อยู่ จะเกิดข้อผิดพลาดขึ้น สำหรับข้อมูลเพิ่มเติมเกี่ยวกับข้อผิดพลาดนี้ โปรดดูที่ “ ■ สถานะข้อผิดพลาดของการ์ด CF” (หน้า 21-40) |

## ตัวอย่างนิพจน์

```
[w:[#INTERNAL]LS0200] = 0 // Offset (“0” when the mode is “New”)
```

```
[w:[#INTERNAL]LS0202] = 100 // No. of Bytes (100 bytes)
```

```
[w:[#INTERNAL]LS0204] = 0 // Mode (New)
```

```
_CF_write (“\DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100],
```

```
[w:[#INTERNAL]LS0200], [w:[#INTERNAL]LS0202], [w:[#INTERNAL]:LS0204])
```

ตัวอย่างข้างบนเป็นการสร้างไฟล์ใหม่ชื่อ DATA0001.BIN ไว้ในโฟลเดอร์ \DATA และจัดเก็บข้อมูลจำนวน 100 ไบต์ที่อ่านจากพื้นที่ LS0100

เมื่อระบุอุปกรณ์ภายในให้กับออฟเซต จำนวนไบต์ หรือโหมด คุณสามารถกำหนดออฟเซต จำนวนไบต์ หรือโหมดโดยทางอ้อมได้

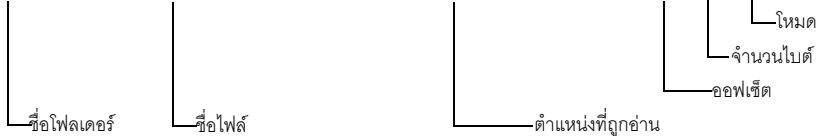
## ข้อสำคัญ

- การตั้งค่าออฟเซตจะมีผลเฉพาะในโหมด “Overwrite” เท่านั้น ส่วนในโหมด “New” และโหมด “Add” จะตั้งค่าออฟเซตไม่ได้ ให้ตั้งค่าออฟเซตเป็น “0” ในโหมดอื่นที่ไม่ใช่โหมด “Overwrite”
- เมื่อระบุโหมด “New” และมีไฟล์ที่มีชื่อเหมือนกันอยู่แล้ว ไฟล์นั้นจะถูกเขียนทับ
- เมื่อระบุพื้นที่ LS ไว้สำหรับ “ชื่อไฟล์” แล้ว “ตำแหน่งที่ถูกอ่าน” จะไม่นับเป็นตำแหน่งของ D-Script
- เมื่อระบุอุปกรณ์ PLC ไว้สำหรับ “ตำแหน่งที่ถูกอ่าน” เมื่อมีการเรียกใช้ฟังก์ชัน ระบบจะอ่านข้อมูลจาก PLC เพียงครั้งเดียวเท่านั้น หากเกิดข้อผิดพลาดระหว่างการอ่านข้อมูล ระบบจะกำหนดข้อผิดพลาดไว้ในสถานะข้อผิดพลาดของการ์ด CF [s:CF\_ERR\_STAT] และจะล้างข้อผิดพลาดออกเมื่ออ่านข้อมูลเสร็จสมบูรณ์
- ระบบจะแบ่งข้อมูลออกเป็นส่วนๆ และอ่านจากต้นฉบับ แม้ว่าจะขึ้นอยู่กับจำนวนไบต์ที่จะอ่านด้วยก็ตาม ดังนั้น ถึงแม้จะเกิดข้อผิดพลาดในการสื่อสารขึ้นระหว่างที่อ่านข้อมูล แต่อาจมีข้อมูลบางส่วนถูกเขียนลงในไฟล์ที่ระบุแล้ว
- หากต้องการระบุพาธแบบครบถ้วนเป็นชื่อไฟล์ ให้ระบุ “\*” (เครื่องหมายดอกจัน) เป็นชื่อโฟลเดอร์ เช่น \_CF\_read (“\*”, “\DATA\DATA0001.BIN”, [w:[#INTERNAL]LS0100], 0, 10)

## ตัวอย่างนิพจน์ของรูปแบบการจัดเก็บข้อมูล

## ◆ เมื่อระบุโหมด “New”

```
_CF_write (“\DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 0, 100, 0 )
```



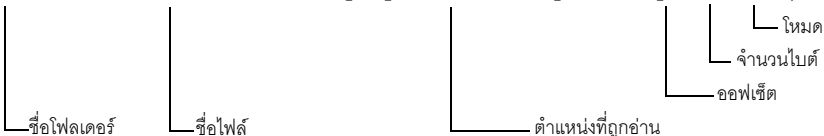
เมื่อมีการเรียกใช้ข้อความคำสั่งข้างบน ระบบจะอ่านข้อมูลจำนวน 100 ไบต์จากพื้นที่ LS0100 และพื้นที่ถัดไป แล้วเขียนลงในไฟล์ DATA0001.BIN ที่สร้างขึ้นใหม่ในโฟลเดอร์ \DATA

## ข้อสำคัญ

- สามารถใช้ชื่อไฟล์ได้เฉพาะรูปแบบ 8.3 เท่านั้น (รวมทั้งหมดไม่เกิน 12 อักขระ โดยแบ่งเป็นชื่อไฟล์ 8 อักขระและนามสกุลอีก 3 อักขระ) ชื่อไฟล์ที่ยาวกว่ารูปแบบนี้ไม่สามารถใช้ได้

## ◆ เมื่อระบุโหมด “Add”

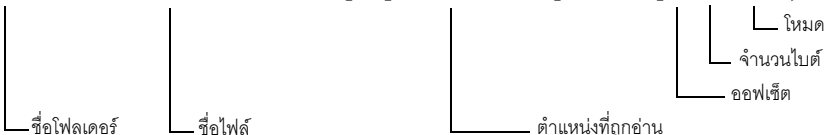
```
_CF_write (“\DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 0, 100, 1 )
```



หากมีไฟล์ที่ระบุอยู่แล้ว (เช่น DATA0001.BIN) และเรียกใช้ข้อความคำสั่งข้างบน ระบบจะอ่านข้อมูลจำนวน 100 ไบต์จากพื้นที่ LS0100 และพื้นที่ถัดไป แล้วเพิ่มลงในไฟล์ DATA0001.BIN ในโฟลเดอร์ \DATA

## ◆ เมื่อระบุโหมด “Overwrite” (1)

```
_CF_write (“\DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 16, 10, 2 )
```

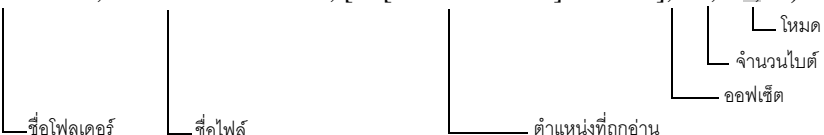


หากมีไฟล์ที่ระบุอยู่แล้ว (เช่น DATA0001.BIN) และเรียกใช้ข้อความคำสั่งข้างบน ระบบจะอ่านข้อมูลจำนวน 10 ไบต์ที่จัดเก็บไว้ในพื้นที่ LS0100 และพื้นที่ถัดไป แล้วเขียนทับข้อมูลจำนวน 10 ไบต์ที่จัดเก็บอยู่ในไบต์ที่ 17 และไบต์ถัดไปต่อจากออฟเซตในไฟล์ DATA0001.BIN ในโฟลเดอร์ \DATA

## ◆ เมื่อระบุโหมด “Overwrite” (2)

(ไฟล์ที่จะเขียนทับมีน้อยกว่ายอดรวมของค่าออฟเซตและจำนวนไบต์)

```
_CF_write (“\DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 96, 10, 2 )
```

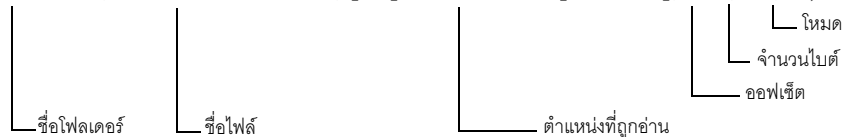


มีไฟล์ที่ระบุอยู่แล้ว (เช่น DATA0001.BIN) และขนาดไฟล์เท่ากับ 100 ไบต์ เมื่อตั้งค่าออฟเซตเป็น 96 ไบต์ และตั้งค่าจำนวนไบต์สำหรับการเขียนทับไว้ 10 ไบต์ ระบบจะอ่านข้อมูลจำนวน 10 ไบต์ที่จัดเก็บอยู่ในพื้นที่ LS0100 และพื้นที่ถัดไป จากนั้น ข้อมูลที่อ่านได้ 4 ไบต์แรกจะเขียนทับข้อมูล 4 ไบต์ที่จัดเก็บไว้ในไบต์ที่ 97 และไบต์ถัดไปในไฟล์ ส่วนข้อมูลที่เหลืออีก 6 ไบต์จะถูกเพิ่มลงในตอนท้ายของข้อมูลในไฟล์ ไฟล์ผลลัพธ์ที่ได้ จะมีข้อมูลเท่ากับ 106 ไบต์

◆ เมื่อระบุโหมด “Overwrite” (3)

(ไฟล์ที่จะถูกเขียนทับเล็กกว่าค่าออฟเซต)

\_CF\_write (“\DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 110, 10, 2 )



มีไฟล์ที่ระบุอยู่แล้ว (เช่น DATA0001.BIN) และขนาดไฟล์เท่ากับ 100 ไบต์ เมื่อตั้งค่าออฟเซตเป็น 110 ไบต์ และตั้งค่าจำนวนไบต์สำหรับการเขียนทับไว้ 10 ไบต์ ระบบจะเติม 0 ลงในพื้นที่ระหว่างไบต์ที่ 101 และไบต์ที่ 110 และเขียนข้อมูลจำนวน 10 ไบต์ที่อ่านจากพื้นที่ LS0100 และพื้นที่ถัดไปลงไบต์ที่ 111 และไบต์ถัดไป ไฟล์ผลลัพธ์ที่ได้จะมีข้อมูลเท่ากับ 120 ไบต์

ข้อสำคัญ

- พารามิเตอร์แรก “ชื่อไฟล์เดอร์” และพารามิเตอร์ที่สอง “ชื่อไฟล์” สามารถใช้อักขระแบบไบต์เดียวได้สูงสุด 32 ตัว
- สามารถระบุอุปกรณ์ภายในเป็น “ชื่อไฟล์” ของพารามิเตอร์ที่สองได้ การระบุอุปกรณ์ภายในทำให้สามารถอ้างตำแหน่งของชื่อไฟล์โดยอ้อมได้ โดยสามารถระบุชื่อไฟล์ด้วยอักขระแบบไบต์เดียวได้สูงสุด 32 ตัว

เช่น \_CF\_write (“\DATA”, [w:LS0100], [w:[#INTERNAL]LS0200], 0, 100, 0)

การจัดเก็บชื่อไฟล์ในพื้นที่ LS0100 ช่วยอ้างตำแหน่งชื่อไฟล์โดยอ้อมได้ ในตัวอย่างนี้ชื่อไฟล์จะถูกจัดเก็บไว้ในพื้นที่ LS0100 จนถึง LS0106 ดังต่อไปนี้

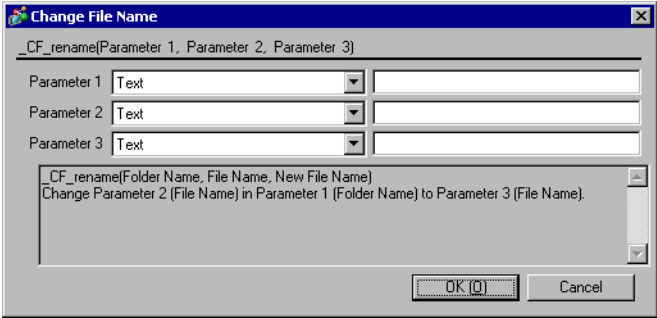
|        |        |      |
|--------|--------|------|
|        | 16 บิต |      |
| LS0100 | 'D'    | 'A'  |
| LS0101 | 'T'    | 'A'  |
| LS0102 | '0'    | '0'  |
| LS0103 | '0'    | '1'  |
| LS0104 | ':'    | 'B'  |
| LS0105 | 'I'    | 'N'  |
| LS0106 | '\0'   | '\0' |
|        | :      |      |

ตอนท้ายของชื่อไฟล์ต้องเป็นอักขระค่าศูนย์  
อุปกรณ์แสดงผลรับข้อมูลที่อยู่ก่อนหน้าอักขระ  
ค่าศูนย์ว่าเป็นชื่อไฟล์

จากตัวอย่างข้างบน ระบบจะอ่านข้อมูลจำนวน 100 ไบต์จากพื้นที่ LS0200 และพื้นที่ถัดไป และสร้างไฟล์ใหม่ชื่อ “\DATA\DATA0001.BIN” เพื่อใช้จัดเก็บข้อมูล

- ชื่อไฟล์สามารถใช้ได้เฉพาะ “รูปแบบ 8.3” เท่านั้น (อักขระสูงสุด 12 อักขระ โดยเป็นชื่อไฟล์ 8 อักขระและนามสกุลอีก 3 อักขระ) ไม่สามารถใช้ชื่อไฟล์ยาวกว่านี้ได้

## 21.5.3 Change File Name

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | แก้ไขชื่อไฟล์ Parameter 1 จะกำหนดโฟลเดอร์ข้อมูลการ์ด CF Parameter 2 จะกำหนดชื่อไฟล์ต้นฉบับ Parameter 3 จะกำหนดชื่อใหม่                                                                                                                                                                                                                                                                                                                                                                                                |
| รูปแบบ     | <p>_CF_rename (ชื่อโฟลเดอร์, ชื่อไฟล์, ชื่อไฟล์ใหม่)<br/>         คุณสามารถกำหนดชื่อไฟล์โดยอ้อมด้วยตำแหน่ง LS ได้ด้วย</p>  <p>Parameter 1<br/>         ชื่อโฟลเดอร์: ข้อความที่กำหนดไว้ตายตัว</p> <p>Parameter 2<br/>         ชื่อไฟล์: ข้อความที่กำหนดไว้ตายตัว, อุปกรณ์ภายใน, อุปกรณ์ภายใน + ตำแหน่งชั่วคราว</p> <p>Parameter 3<br/>         ชื่อไฟล์: ข้อความที่กำหนดไว้ตายตัว, อุปกรณ์ภายใน, อุปกรณ์ภายใน + ตำแหน่งชั่วคราว</p> |

## ตัวอย่างนิพจน์

\_CF\_rename (“DATA”, “DATA0001.BIN”, “DATA1234.BIN”)

ตัวอย่างข้างบนเป็นการเปลี่ยนชื่อไฟล์จาก “DATA\DATA0001.BIN” เป็น “DATA\DATA1234.BIN”

### ข้อสำคัญ

- ชื่อไฟล์สามารถใช้ได้เฉพาะ “รูปแบบ 8.3” เท่านั้น (อักขระสูงสุด 12 อักขระ โดยเป็นชื่อไฟล์ 8 อักขระและนามสกุลอีก 3 อักขระ) ไม่สามารถใช้ชื่อไฟล์ยาวกว่านี้ได้
- พารามิเตอร์แรก “ชื่อไฟล์เดอรั” และพารามิเตอร์ที่สอง “ชื่อไฟล์” สามารถใช้อักขระแบบไบต์เดียวได้สูงสุด 32 ตัว
- สามารถระบุอุปกรณ์ภายในเป็น “ชื่อไฟล์” ของพารามิเตอร์ที่สองหรือสามได้ การระบุอุปกรณ์ภายในทำให้สามารถอ้างตำแหน่งของชื่อไฟล์โดยอ้อมได้ โดยสามารถระบุชื่อไฟล์ด้วยอักขระแบบไบต์เดียวได้สูงสุด 32 ตัว

ตัวอย่าง

\_CF\_rename (“DATA”, [w:[#INTERNAL]LS0100],

[w:[#INTERNAL]LS0200])

การจัดเก็บชื่อไฟล์ในพื้นที่ LS0100 และ LS0200 ทำให้อ้างตำแหน่งชื่อไฟล์ได้โดยอ้อม

- จัดเก็บชื่อไฟล์ในพื้นที่ LS0100 จนถึง LS0106 ดังต่อไปนี้

16 บิต

|        |      |      |
|--------|------|------|
| LS0100 | 'D'  | 'A'  |
| LS0101 | 'T'  | 'A'  |
| LS0102 | '0'  | '0'  |
| LS0103 | '0'  | '1'  |
| LS0104 | '.'  | 'B'  |
| LS0105 | 'I'  | 'N'  |
| LS0106 | '\0' | '\0' |

ตอนท้ายของชื่อไฟล์ต้องเป็นอักขระค่าศูนย์  
GP รับรู้ข้อมูลที่อยู่ก่อนหน้าอักขระค่าศูนย์ว่า  
เป็นชื่อไฟล์

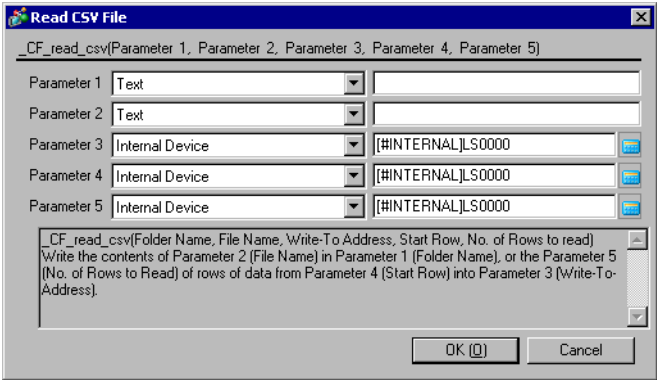
16 บิต

|        |      |      |
|--------|------|------|
| LS0200 | 'D'  | 'A'  |
| LS0201 | 'T'  | 'A'  |
| LS0202 | '1'  | '2'  |
| LS0203 | '3'  | '4'  |
| LS0204 | '.'  | 'B'  |
| LS0205 | 'I'  | 'N'  |
| LS0206 | '\0' | '\0' |

จากตัวอย่างข้างบน ชื่อของไฟล์ “DATA\DATA0001.BIN” ถูกเปลี่ยนเป็น “DATA\DATA1234.BIN”

- เมื่อระบุพื้นที่ LS เป็น “ชื่อไฟล์” พื้นที่นั้นจะไม่นับเป็นตำแหน่งของ D-Script
- หากต้องการระบุไฟล์เดอรั (ไดเรกทอรี) ให้ระบุชื่อไฟล์เดอรัเป็น “ ” (สตริงเปล่า)
- หากต้องการระบุพารามิเตอร์เป็นชื่อไฟล์ ให้ระบุ “\*” (เครื่องหมายดอกจัน) เป็นชื่อไฟล์เดอรั

## 21.5.4 Read CSV File

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | อ่านข้อมูลในเซลล์จากไฟล์ CSV (ซึ่งสร้างจากภาพเซลล์ที่คั่นด้วย “,”) และเขียนลงในตำแหน่งเวิร์ด                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| รูปแบบ     | <p>_CF_read_csv (ชื่อโฟลเดอร์, ชื่อไฟล์, [ตำแหน่งปลายทางการเขียน], แถวเริ่มต้น, จำนวนแถวที่จะอ่าน)</p>  <p>Parameter 1: ข้อความ (อักขระแบบไบต์เดี่ยวสูงสุด 32 อักขระ)</p> <p>Parameter 2: ข้อความ (อักขระแบบไบต์เดี่ยวสูงสุด 32 อักขระ), อุปกรณ์ภายใน, อุปกรณ์ภายใน + ตำแหน่งชั่วคราว</p> <p>Parameter 3: อุปกรณ์ภายใน, อุปกรณ์ภายในที่กำหนดด้วยออฟเซต</p> <p>Parameter 4: ค่าตัวเลข (1 ถึง 65,535), อุปกรณ์ภายใน, ตัวแปรชั่วคราว</p> <p>Parameter 5: ค่าตัวเลข (1 ถึง 65,535), อุปกรณ์ภายใน, ตัวแปรชั่วคราว</p> |

## ตัวอย่างนิพจน์

\_CF\_read\_csv (“\CSV”, “SAMPLE.CSV”, [w:[#INTERNAL]LS1000], 1, 2)  
 (เมื่อใช้ฟังก์ชัน “\_CF\_read\_csv ( )” อ่านข้อมูลในการ์ดหน่วยความจำ CF จำนวนสองบรรทัด เริ่มจากบรรทัดแรก  
 ของไฟล์ [CSV\SAMPLE.CSV])

### SAMPLE.CSV

|                          |   |
|--------------------------|---|
| 001,"DAT01-01","DAT01-2" | ← |
| 002,"DAT02-01","DAT02-2" | ← |

อ่านข้อมูลของไฟล์ CSV จำนวนสองบรรทัดโดยเริ่มจากบรรทัดแรก  
 เมื่ออักขระตัวแรกเป็นค่าตัวเลข (“0” ถึง “9”, “\_”, “.”), ข้อมูลจะถูก  
 จัดเก็บเป็นค่าตัวเลข เมื่ออักขระตัวแรกเป็น ["] ระบบจะถือว่า  
 ข้อมูลเป็นอักขระและจัดเก็บค่า “00h” ไว้ที่ตอนท้ายของสตริง  
 ข้อความ เช่น เมื่อจัดเก็บ “DAT01-01” ขนาดข้อมูลคือ 8 อักขระ  
 ซึ่งเป็นจำนวนคู่ และใช้เวร็ดทั้งหมด 5 เวร็ด โดย 4 เวร็ดใช้สำหรับ  
 จัดเก็บสตริงข้อความ และอีก 1 เวร็ดสำหรับจัดเก็บ “00h” ไว้ตอนท้าย  
 เช่น เมื่อจัดเก็บ “DAT01-2” ขนาดข้อมูลคือ 7 อักขระซึ่งเป็นจำนวนคี่  
 และใช้เวร็ดทั้งหมด 4 เวร็ดเพื่อจัดเก็บข้อความโดยจะจัดเก็บ “00h”  
 ไว้ตอนท้าย

| 16 บิต |         |
|--------|---------|
| LS1000 | 1       |
| +1     | 'D' 'A' |
| +2     | 'T' '0' |
| +3     | '1' '.' |
| +4     | '0' '1' |
| +5     | 00h 00h |
| +6     | 'D' 'A' |
| +7     | 'T' '0' |
| +8     | '1' '.' |
| +9     | '2' 00h |
| LS1010 | 2       |
| +1     | 'D' 'A' |
| +2     | 'T' '0' |
| +3     | '2' '.' |
| +4     | '0' '1' |
| +5     | 00h 00h |
| +6     | 'D' 'A' |
| +7     | 'T' '0' |
| +8     | '2' '.' |
| +9     | '2' 00h |

เมื่อโหมดการจัดเก็บข้อมูลคือโหมด 0

### หมายเหตุ

- เมื่ออักขระตัวแรกในเซลล์เป็นค่าตัวเลข (“0” ถึง “9”, “\_”, “.”) ระบบจะแปลงค่าเป็นข้อมูลตัวเลข แล้วเขียนข้อมูลนั้นลงในอุปกรณ์ LS ช่วงที่สามารถใช้ได้คือตั้งแต่ -32,768 ถึง 32,767
- เมื่ออักขระแรกในเซลล์เป็น ["] ระบบจะเขียนช่วงที่มีอักขระ ["] ลงในอุปกรณ์ LS เป็นข้อมูลสตริงข้อความ เมื่อจำนวนไบต์ของข้อมูลสตริงข้อความเป็นเลขคี่ ระบบจะเติม “0x00” ไว้ตอนท้ายเมื่อจำนวนไบต์ของข้อมูลข้อความเป็นเลขคู่ ระบบจะเขียน “0x0000” ลงในตำแหน่งต่อจากตำแหน่งสุดท้าย ในหนึ่งเซลล์สามารถป้อนอักขระแบบไบต์เดี่ยวได้สูงสุด 32 อักขระ
- เมื่อไฟล์ CSV มีข้อมูล 2 บรรทัดขึ้นไป สามารถอ่านข้อมูลเป็นจำนวนบรรทัดที่ต้องการได้โดยเริ่มจากบรรทัดที่ระบุไว้ ในหนึ่งบรรทัดสามารถป้อนอักขระแบบไบต์เดี่ยวได้สูงสุด 200 อักขระ และในไฟล์ CSV หนึ่งไฟล์สามารถป้อนข้อมูลได้สูงสุด 65,535 บรรทัด
- เมื่อเกิดข้อผิดพลาด ระบบจะเขียนสถานะข้อผิดพลาดลงในพื้นที่ LS9137
- ขณะเขียนข้อมูลข้อความของไฟล์ CSV ลงในอุปกรณ์ LS ลำดับการจัดเก็บข้อมูลจะขึ้นอยู่กับโหมดการจัดเก็บข้อมูล

## สถานะข้อผิดพลาด

|        |            |  |  |
|--------|------------|--|--|
| LS9137 | พื้นที่ LS |  |  |
|        |            |  |  |
|        |            |  |  |

| ชื่อฟังก์ชันตัวแก้ไข | พื้นที่ LS | สถานะข้อผิดพลาด | สาเหตุ                                                                                    |
|----------------------|------------|-----------------|-------------------------------------------------------------------------------------------|
| _CF_read_csv ( )     | LS9137     | 0000h           | เสร็จสมบูรณ์                                                                              |
|                      |            | 0001h           | พารามิเตอร์เกิดข้อผิดพลาด                                                                 |
|                      |            | 000h            | ข้อผิดพลาดของการ์ด CF (ไม่มีการ์ด CF/การเปิดไฟล์เกิดข้อผิดพลาด/การอ่านไฟล์เกิดข้อผิดพลาด) |
|                      |            | 0003h           | การเขียน/การอ่านเกิดข้อผิดพลาด                                                            |

### ข้อสำคัญ

- เมื่อระบุชื่อไฟล์เดอริเป็น “\*” คุณสามารถใช้พารามิเตอร์เป็นชื่อไฟล์ได้
- สามารถใช้ชื่อไฟล์ได้เฉพาะรูปแบบ 8.3 เท่านั้น (รวมทั้งหมดไม่เกิน 12 อักขระ โดยแบ่งเป็นชื่อไฟล์ 8 อักขระและนามสกุลอีก 3 อักขระ) ชื่อไฟล์ที่ยาวกว่ารูปแบบนี้ไม่สามารถใช้ได้
- พื้นที่อุปกรณ์ LS ที่สามารถใช้จัดเก็บข้อมูลที่นำเข้ามาจากไฟล์ CSV ได้ มีเฉพาะพื้นที่สำหรับผู้ใช้ที่กำหนดไว้เท่านั้น (LS20 ถึง LS2031 และ LS2096 ถึง LS8191)
- เวลาที่ต้องใช้ในการประมวลผลสำหรับการนำเข้าข้อมูลจะเป็นสัดส่วนกับปริมาณข้อมูลของไฟล์ CSV ที่จะอ่าน ระบบจะไม่รีเฟรชพาร์ทจนกว่าจะประมวลผลเสร็จสมบูรณ์ (การอ่านข้อมูลในไฟล์ CSV ซึ่งมี 100 บรรทัด บรรทัดละ 40 อักขระ ตั้งแต่บรรทัดแรกจนถึงบรรทัดที่ 100 จะใช้เวลาประมาณ 10 วินาที)
- ฟังก์ชันจะไม่ถูกบันทึกลงใน [s:CF\_ERR\_STAT] ในทันทีหลังจากเรียกใช้ฟังก์ชัน ซึ่งจะแตกต่างจากฟังก์ชัน [\_CF\_read ( )] (ในบางกรณี อาจจัดเก็บค่าที่ไม่ได้กำหนดไว้ด้วย)
- สตริงข้อความที่ขึ้นด้วยตัวเลขต้องใส่ [“] ไว้หน้าและหลังสตริงด้วย  
ตัวอย่าง  
[ 123, 2-D4EA ] [ 123, “2-D4EA”]  
ไม่ต้อง                      ถูกต้อง

## 21.5.5 Read File

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | อ่านข้อมูลที่อยู่ต่อจากออฟเซตที่ระบุไว้ตามจำนวนไบต์ที่ระบุ และเขียนข้อมูลที่อ่านได้ลงในตำแหน่งปลายทาง โปรดดูรายละเอียดเพิ่มเติมเกี่ยวกับลำดับการจัดเก็บข้อมูลได้ในส่วน “โหมดการจัดเก็บข้อมูล” ข้างล่าง                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| รูปแบบ     | <p>_CF_read (ชื่อโฟลเดอร์, ชื่อไฟล์, ตำแหน่งปลายทางการเขียน, ออฟเซต, จำนวนไบต์)</p> <p>Parameter 1<br/>ชื่อโฟลเดอร์: ข้อความที่กำหนดไว้ตายตัว (ความยาวสูงสุด: อักขระแบบไบต์เดี่ยว 32 อักขระ)</p> <p>Parameter 2<br/>ชื่อไฟล์: สตริงที่กำหนดไว้ตายตัว, อุปกรณ์ภายใน, อุปกรณ์ภายใน + ตำแหน่งชั่วคราว (ความยาวสูงสุด: อักขระแบบไบต์เดี่ยว 32 ตัว)</p> <p>Parameter 3<br/>ตำแหน่งปลายทางการเขียน: ตำแหน่งอุปกรณ์, ตำแหน่งอุปกรณ์ + ตำแหน่งชั่วคราว</p> <p>Parameter 4<br/>ออฟเซต: ค่าตัวเลข, ตำแหน่งอุปกรณ์, ตำแหน่งชั่วคราว (จำนวนสูงสุดที่สามารถระบุได้: 65,535 สำหรับความยาว 16 บิต, 4,294,967,295 สำหรับความยาว 32 บิต)</p> <p>Parameter 5<br/>จำนวนไบต์: ค่าตัวเลข, ตำแหน่งอุปกรณ์, ตำแหน่งชั่วคราว (จำนวนสูงสุดที่สามารถระบุได้: 1,280)</p> |

## ตัวอย่างนิพจน์

การอ่านข้อมูลจำนวน 16 ไบต์ในไฟล์ที่ระบุ เมื่อออฟเซตเท่ากับ 16

\_CF\_read (“\DATA”, “DATA0001.BIN”, [w:[#INTERNAL]LS0100], 16, 16)

จากตัวอย่างข้างบน ระบบจะเขียนข้อมูลจำนวน 16 ไบต์จากไบต์ที่ 17 เป็นต้นไปในไฟล์ “\DATA\DATA0001.BIN” ลงในพื้นที่ LS0100 และพื้นที่ถัดไป

**ข้อสำคัญ**

- ชื่อไฟล์สามารถใช้ได้เฉพาะ “รูปแบบ 8.3” เท่านั้น (อักขระสูงสุด 12 อักขระ โดยเป็นชื่อไฟล์ 8 อักขระและนามสกุลอีก 3 อักขระ) ไม่สามารถใช้ชื่อไฟล์ยาวกว่านี้ได้
- พารามิเตอร์แรก “ชื่อไฟล์เดอริ” และพารามิเตอร์ที่สอง “ชื่อไฟล์” สามารถใช้อักขระแบบไบต์เดียวได้สูงสุด 32 ตัว
- สามารถระบุอุปกรณ์ภายในเป็น “ชื่อไฟล์” ของพารามิเตอร์ที่สองได้ การระบุอุปกรณ์ภายในทำให้สามารถอ้างตำแหน่งของชื่อไฟล์โดยอ้อมได้ โดยสามารถระบุชื่อไฟล์ด้วยอักขระแบบไบต์เดียวได้สูงสุด 32 ตัว

ตัวอย่าง

การอ่านข้อมูลจำนวน 10 ไบต์ที่จัดเก็บไว้ในไฟล์ เมื่อระบุไฟล์ในพื้นที่ LS0100 และพื้นที่ถัดไป และค่าออฟเซตเท่ากับ 0

\_CF\_read (“DATA”, [w:LS0100], [w:LS0200], 0, 10)

การจัดเก็บชื่อไฟล์ในพื้นที่ LS0100 ช่วยอ้างตำแหน่งชื่อไฟล์โดยอ้อมได้ ในตัวอย่างนี้ชื่อไฟล์จะถูกจัดเก็บไว้ในพื้นที่ LS0100 จนถึง LS0106 ดังต่อไปนี้

|        |        |      |
|--------|--------|------|
|        | 16 บิต |      |
| LS0100 | 'D'    | 'A'  |
| LS0101 | 'T'    | 'A'  |
| LS0102 | 'O'    | 'O'  |
| LS0103 | 'O'    | '1'  |
| LS0104 | '.'    | 'B'  |
| LS0105 | 'I'    | 'N'  |
| LS0106 | '\0'   | '\0' |
|        | :      | :    |

ตอนท้ายของชื่อไฟล์ต้องเป็นอักขระค่าศูนย์  
อุปกรณ์แสดงผลลัพธ์ข้อมูลที่อยู่ก่อนหน้าอักขระ  
ค่าศูนย์ว่าเป็นชื่อไฟล์

จากตัวอย่างข้างบน ระบบจะอ่านข้อมูลจำนวน 10 ไบต์จากจุดเริ่มต้นของไฟล์ “DATA\DATA0001.BIN” แล้วเขียนลงใน LS0200 และพื้นที่ถัดไป

- จำนวนไบต์ที่อ่านได้สำเร็จจะถูกเขียนลงในจำนวนไบต์ที่อ่านของการ์ด CF [s:CF\_READ\_NUM] สำหรับรายละเอียดเพิ่มเติม โปรดดูที่ “21.5.1 Label Settings ■ สถานะข้อผิดพลาดของการ์ด CF” (หน้า 21-40)
- อุปกรณ์ภายในที่ระบุใน “ชื่อไฟล์” และ “ตำแหน่งปลายทางการเขียน” จะไม่นับเป็นตำแหน่งของ D-Script
- เมื่อระบุตำแหน่งปลายทางการเขียนเป็นอุปกรณ์ PLC จำเป็นต้องใช้เวลาเพิ่มขึ้นในการเขียนข้อมูลลงใน PLC เพราะจำนวนเวิร์ด (ไบต์) เพิ่มขึ้น โดยอาจต้องใช้เวลามากกว่าวินาทีขึ้นอยู่กับจำนวนเวิร์ด
- หากข้อมูลที่อ่านจากไฟล์เกินกว่าช่วงอุปกรณ์ที่กำหนดไว้ของ PLC จะเกิดข้อผิดพลาดในการสื่อสารขึ้น ในกรณีนี้ คุณต้องปิดเครื่อง PLC แล้วเปิดใหม่เพื่อให้ PLC หลบจากข้อผิดพลาด
- เมื่อระบุปลายทางเป็นอุปกรณ์ PLC ค่าจะไม่ถูกเขียนในทันทีเนื่องจากต้องใช้เวลาในการส่งข้อมูลจาก GP ไปยัง PLC

ตัวอย่าง

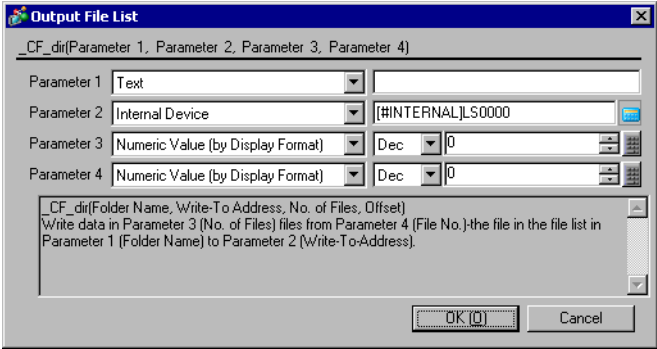
ในสคริปต์ด้านล่าง ข้อความคำสั่ง (1) จะอ่านข้อมูลจำนวน 10 ไบต์จากไฟล์และเขียนข้อมูลลงใน [w:D0100] เมื่อเรียกใช้ข้อความคำสั่ง (2) ระบบยังไม่ได้เขียนข้อมูลลงใน [w:[PLC1]D0100] แต่อย่างใด เพราะต้องใช้เวลาในการส่งข้อมูล

\_CF\_read (“DATA”, “DATA0001.BIN”, [w:[PLC1]D0100], 0, 10) ..... (1)  
[w:[PLC1]D0200] = [w:[PLC1]D0100] + 1 ..... (2)

ในกรณีเช่นนี้ ให้จัดเก็บข้อมูลหนึ่งครั้งในพื้นที่ LS แล้วเรียกใช้ข้อความคำสั่งที่สองดังต่อไปนี้

\_CF\_read (“DATA”, “DATA0001.BIN”, [w:[PLC1]D0100], 0, 10)  
memcpy ([w:[#INTERNAL]LS0100], [w:[PLC1]D0100], 10)  
[w:[PLC1]D0200] = [w:[#INTERNAL]LS0100] + 1

## 21.5.6 Output File List

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | เขียนรายการไฟล์ที่มีอยู่ในโฟลเดอร์ที่ระบุลงในอุปกรณ์ภายใน Parameter 1 จะกำหนดโฟลเดอร์ข้อมูลการ์ด CF Parameter 4 จะกำหนดค่าออฟเซตที่ใช้เลือกไฟล์หนึ่งหรือหลายไฟล์ในโฟลเดอร์นั้น Parameter 3 จะกำหนดจำนวนไฟล์ที่เลือกภายในโฟลเดอร์นั้น Parameter 2 จะระบุพื้นที่ LS ลงในไฟล์ที่จะถูกเขียน เมื่อระบุออฟเซตเป็น “0” รายการจะเริ่มจากไฟล์ (เริ่มต้น) แรกสุด                                                                                                                                                                                                   |
| รูปแบบ     | <p>_CF_dir (ชื่อโฟลเดอร์, ตำแหน่งปลายทางการเขียน, จำนวนไฟล์, ออฟเซต)</p>  <p>Parameter 1<br/>ชื่อโฟลเดอร์: ข้อความที่กำหนดไว้ตายตัว (ความยาวสูงสุด: อักขระแบบไบต์เดี่ยว 32 อักขระ)</p> <p>Parameter 2<br/>ตำแหน่งปลายทางการเขียน: อุปกรณ์ภายใน, อุปกรณ์ภายในที่กำหนดด้วยออฟเซต</p> <p>Parameter 3<br/>จำนวน ไฟล์: ค่าตัวเลข, ตำแหน่งอุปกรณ์, ตำแหน่งชั่วคราว (จำนวนสูงสุดที่ระบุได้: 32)</p> <p>Parameter 4<br/>ออฟเซต: ค่าตัวเลข, ตำแหน่งอุปกรณ์, ตำแหน่งชั่วคราว</p> |

### ตัวอย่างนิพจน์

การส่งข้อมูลรายการไฟล์ที่มีสองไฟล์เมื่อค่าออฟเซตเท่ากับ 1 (ไฟล์ที่สอง)

\_CF\_dir (“\DATA\\*.\*”, [w:[#INTERNAL]LS0100], 2, 1)

เมื่อเรียกใช้ข้อความคำสั่งข้างบนขณะมีไฟล์ต่อไปนี้อยู่ในโฟลเดอร์ DATA ชื่อไฟล์ “DATA0001.BIN” และ “DATA02.BIN” จะถูกเขียนลงในพื้นที่ LS0100 และพื้นที่ถัดไป

| รายละเอียดของโฟลเดอร์ |              | รายละเอียดของพื้นที่ LS |           |
|-----------------------|--------------|-------------------------|-----------|
|                       |              | 16 บิต                  |           |
| /DATA                 | DATA0000.BIN | LS0100                  | 'D' 'A'   |
|                       | DATA0001.BIN | LS0101                  | 'T' 'A'   |
|                       | DATA02.BIN   | LS0102                  | '0' '0'   |
|                       | DATA0003.BIN | LS0103                  | '0' '1'   |
|                       | DATA0004.BIN | LS0104                  | '.' 'B'   |
|                       |              | LS0105                  | 'I' 'N'   |
|                       |              | LS0106                  | '\0' '\0' |
|                       |              | LS0107                  | 'D' 'A'   |
|                       |              | LS0108                  | 'T' 'A'   |
|                       |              | LS0109                  | '0' '2'   |
|                       |              | LS0110                  | '.' 'B'   |
|                       |              | LS0111                  | 'I' 'N'   |
|                       |              | LS0112                  | '\0' '\0' |
|                       |              | LS0113                  | '\0' '\0' |

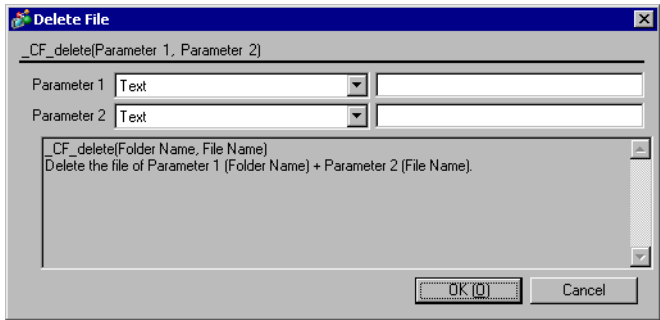
ใช้เวร็ด 7 เวิร์ด

ใช้เวร็ด 7 เวิร์ด

ข้อสำคัญ

- เมื่อระบุออฟเซตเป็น “0” รายการจะเริ่มจากไฟล์ (เริ่มต้น) แรกสุด
- ชื่อไฟล์สามารถใช้ได้เฉพาะ “รูปแบบ 8.3” เท่านั้น (อักขระสูงสุด 12 อักขระ โดยเป็นชื่อไฟล์ 8 อักขระและนามสกุลอีก 3 อักขระ) ไม่สามารถใช้ชื่อไฟล์ยาวกว่านี้ได้
- หากไฟล์เดอที่ระบุมีไฟล์ไม่ครบตามที่ระบุไว้ ระบบจะเติมอักขระที่มีค่าศูนย์ (“0”) ลงในพื้นที่ LS ที่เหลืออยู่
- หากชื่อไฟล์มีน้อยกว่า 12 อักขระ ระบบจะเติมอักขระที่มีค่าศูนย์ (“0”) ลงในตำแหน่งที่ว่างอยู่
- หากต้องการระบุชื่อไฟล์เดอ ต้องพิมพ์ “\*.\*” ลงไปด้วย เช่น “\DATA\\*.\*”  
“\*.\*” หมายความว่าให้แสดงรายการไฟล์ทั้งหมด
- จำนวนไฟล์ตามที่แสดงจริงจะถูกเขียนลงในจำนวนไฟล์ที่แสดงในการ์ด CF [s:CF\_FILELIST\_NUM] สำหรับรายละเอียดเพิ่มเติม โปรดดูที่ “■ สถานะข้อผิดพลาดของการ์ด CF” (หน้า 21-40)
- ตำแหน่ง LS ที่เป็นปลายทางในการเขียนจะไม่เป็นตำแหน่งของ D-Script
- ระบบจะเขียนชื่อไฟล์ลงในพื้นที่ LS โดยไม่เรียงลำดับตัวอักษร แต่จะเขียนตามลำดับการสร้าง (ลำดับการป้อนข้อมูล FAT)
- คุณสามารถสร้างรายการได้โดยระบุนามสกุลไฟล์ หากต้องการแสดงไฟล์เฉพาะที่มีนามสกุลไฟล์ที่ต้องการ ให้ใช้รูปแบบต่อไปนี้ เช่น “\DATA\\*.BIN” แต่ทั้งนี้ ในชื่อไฟล์จะมีเครื่องหมาย “\*” อยู่ไม่ได้

## 21.5.7 Delete File

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                    |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ลบไฟล์ที่ระบุออกจากการ์ด CF Parameter 1 จะกำหนดไฟล์เดอข้อมูลการ์ด CF Parameter 2 จะกำหนดชื่อของไฟล์ที่จะถูกลบ                                                                                                                                                                                                                                               |
| รูปแบบ     | <p>_CF_delete (ชื่อไฟล์เดอ, ชื่อไฟล์)<br/>คุณสามารถกำหนดชื่อไฟล์โดยอ้อมด้วยตำแหน่ง LS ได้ด้วย</p>  <p>Parameter 1<br/>ชื่อไฟล์เดอ: ข้อความที่กำหนดไว้ตายตัว</p> <p>Parameter 2<br/>ชื่อไฟล์: ข้อความที่กำหนดไว้ตายตัว, อุปกรณ์ภายใน, อุปกรณ์ภายใน + ตำแหน่งชั่วคราว</p> |

## ตัวอย่างนิพจน์

\_CF\_delete (“\DATA”, “DATA0001.BIN”)

ตัวอย่างข้างบนเป็นการลบไฟล์ “\DATA \DATA0001.BIN”

## ข้อสำคัญ

- ชื่อไฟล์สามารถใช้ได้เฉพาะ “รูปแบบ 8.3” เท่านั้น (อักขระสูงสุด 12 อักขระ โดยเป็นชื่อไฟล์ 8 อักขระและนามสกุลอีก 3 อักขระ) ไม่สามารถใช้ชื่อไฟล์ยาวกว่านี้ได้
- พารามิเตอร์แรก “ชื่อไฟล์เดอรั” และพารามิเตอร์ที่สอง “ชื่อไฟล์” สามารถใช้อักขระแบบไบต์เดียวได้สูงสุด 32 ตัว
- สามารถระบุอุปกรณ์ภายในเป็น “ชื่อไฟล์” ของพารามิเตอร์ที่สองได้ การระบุอุปกรณ์ภายในทำให้สามารถอ้างตำแหน่งของชื่อไฟล์โดยอ้อมได้ โดยสามารถระบุชื่อไฟล์ด้วยอักขระแบบไบต์เดียวได้สูงสุด 32 ตัว

ในตัวอย่างนี้ ชื่อไฟล์จะถูกจัดเก็บไว้ในพื้นที่ LS0100 จนถึง LS0106 ดังต่อไปนี้

16 บิต

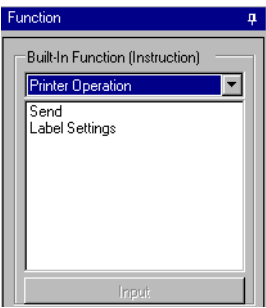
|        |      |      |
|--------|------|------|
| LS0100 | 'D'  | 'A'  |
| LS0101 | 'T'  | 'A'  |
| LS0102 | '0'  | '0'  |
| LS0103 | '0'  | '1'  |
| LS0104 | ':'  | 'B'  |
| LS0105 | 'I'  | 'N'  |
| LS0106 | '\0' | '\0' |

ตอนท้ายของชื่อไฟล์ต้องเป็นอักขระค่าศูนย์  
อุปกรณ์แสดงผลรับรู้ข้อมูลที่อยู่ก่อนหน้าอักขระ  
ค่าศูนย์ว่าเป็นชื่อไฟล์

จากตัวอย่างข้างบน ไฟล์ “\DATA\DATA0001.BIN” ถูกลบทิ้ง

- หากต้องการระบุไฟล์เดอรัว (ไดเรกทอรี) ให้ระบุชื่อไฟล์เดอรัเป็น “ ” (สตริงเปล่า)
- เมื่อระบุพื้นที่ LS เป็น “ชื่อไฟล์” จะไม่นับ “ตำแหน่งปลายทางการเขียน” เป็นตำแหน่งของ D-Script
- หากต้องการระบุพารามิเตอร์แบบครบถ้วนเป็นชื่อไฟล์ ให้ระบุ “\*” (เครื่องหมายดอกจัน) เป็นชื่อไฟล์เดอรั

## 21.6 การทำงานของเครื่องพิมพ์

| Printer Operation                                                                 | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Label Settings</p> <p>☞ “21.6.1 Label Settings” (หน้า 21-60)<br/>กำหนดจากตัวแปรควบคุมและตัวแปรสถานะ</p> <hr/> <p>Send</p> <p>☞ “21.6.2 Send” (หน้า 21-62)<br/>ส่งข้อมูลเท่ากับจำนวนไบต์ที่กำหนดไปยังพอร์ต COM</p> |

### ข้อสำคัญ

- พอร์ตที่สามารถใช้ฟังก์ชันการทำงานของเครื่องพิมพ์ได้ ได้แก่ COM1 หรือ USB/PIO (USB-PIO)

### 21.6.1 Label Settings

#### ■ ตัวแปรควบคุม

ตัวแปรควบคุมนี้ (PRN\_CNTRL) ใช้สำหรับล้างข้อมูลในบัฟเฟอร์การส่งข้อมูลและสถานะข้อผิดพลาด  
ตัวแปรควบคุมนี้เป็นแบบเขียนอย่างเดียว

- รายละเอียดของตัวแปรควบคุม (PRN\_CNTRL)

|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| บิต | รายละเอียด                  |
|-----|-----------------------------|
| 15  | สำรอง                       |
| 14  |                             |
| 13  |                             |
| 12  |                             |
| 11  |                             |
| 10  |                             |
| 9   |                             |
| 8   |                             |
| 7   |                             |
| 6   |                             |
| 5   |                             |
| 4   |                             |
| 3   |                             |
| 2   | 1: ล้างข้อผิดพลาด           |
| 1   | สำรอง                       |
| 0   | 1: ล้างบัฟเฟอร์การส่งข้อมูล |

### ข้อสำคัญ

- เมื่อกำหนดเวิร์ด (เมื่อตั้งค่าบิตพร้อมกันสองบิตขึ้นไป) จะทำการประมวลผลตามลำดับดังนี้  
ล้างข้อผิดพลาด  
↓  
ล้างข้อมูลในบัฟเฟอร์การส่งข้อมูล
- บิตที่สำรองไว้อาจจำเป็นต้องใช้ในภายหลัง ดังนั้น ให้กำหนดเฉพาะบิตที่จำเป็นเท่านั้น

## ■ สถานะ

ตัวแปรสถานะนี้ (PRN\_STAT) ใช้สำหรับตรวจสอบการมีอยู่/การขาดหายไปของข้อมูลในบัพเฟอร์การส่งข้อมูล และใช้สำหรับรับสถานะข้อผิดพลาด ตัวแปรสถานะนี้เป็นแบบเขียนอย่างเดียว

- รายละเอียดของตัวแปรสถานะ (PRN\_STAT)

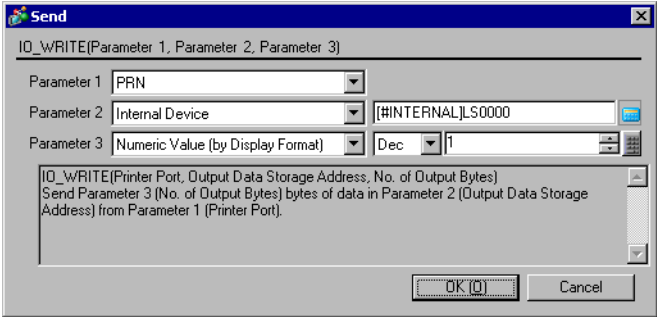
|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| บิต | รายละเอียด                                                                                                  |
|-----|-------------------------------------------------------------------------------------------------------------|
| 15  | สำรอง                                                                                                       |
| 14  | สถานะของสัญญาณ I/F ERROR ของเครื่องพิมพ์<br>ข้อผิดพลาดของเครื่องพิมพ์ (อินพุต):<br>0: ข้อผิดพลาด<br>1: ปกติ |
| 13  | สถานะของสัญญาณ I/F SLCT ของเครื่องพิมพ์<br>เลือก (อินพุต):<br>0: ออฟไลน์<br>1: ออนไลน์                      |
| 12  | สถานะของสัญญาณ I/F PE ของเครื่องพิมพ์<br>กระดาษหมด (อินพุต):<br>0: ปกติ<br>1: กระดาษหมด                     |
| 11  | สำรอง                                                                                                       |
| 10  |                                                                                                             |
| 9   |                                                                                                             |
| 8   |                                                                                                             |
| 7   |                                                                                                             |
| 6   |                                                                                                             |
| 5   |                                                                                                             |
| 4   |                                                                                                             |
| 3   |                                                                                                             |
| 2   |                                                                                                             |
| 1   | 0: ปกติ<br>1: การส่งข้อมูลผิดพลาด                                                                           |
| 0   | 0: มีข้อมูลอยู่ในบัพเฟอร์การส่งข้อมูล<br>1: ไม่มีข้อมูลอยู่ในบัพเฟอร์การส่งข้อมูล                           |

### ข้อสำคัญ

- หากบัพเฟอร์การส่งข้อมูลเกิดโอเวอร์โฟลล์จะเกิดข้อผิดพลาด เมื่อเกิดข้อผิดพลาดนี้ขึ้น บิตข้อผิดพลาดในการส่งจะเปิดขึ้น
- บัพเฟอร์การส่งข้อมูลมีขนาดเท่ากับ 8,192 ไบต์
- บิตที่สำรองไว้อาจถูกกำหนดในภายหลัง ดังนั้น ให้กำหนดเฉพาะบิตที่จำเป็นเท่านั้น

## 21.6.2 Send

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ส่งข้อมูลเท่ากับจำนวนไบต์ที่กำหนดไปยังพอร์ต COM ระบบจะส่งข้อมูลโดยไม่คำนึงถึงชนิดของเครื่องพิมพ์ที่ระบุไว้                                                                                                                                                                                              |
| รูปแบบ     | <p>IO_WRITE ([p:PRN], ตำแหน่งจัดเก็บข้อมูลส่งออก, จำนวนไบต์ที่ส่งออก)</p>  <p>Parameter 1: [p:PRN]<br/>         Parameter 2: อุปกรณ์ภายใน<br/>         Parameter 3: ค่าจำนวนเต็ม, ตำแหน่งอุปกรณ์, ตำแหน่งชั่วคราว</p> |

### ข้อสำคัญ

- ค่าตัวเลขสูงสุดที่สามารถระบุให้ Parameter 3 คือ 1024 ถึงแม้จะระบุค่ามากกว่า 1024 แต่จะส่งออกข้อมูลจากพอร์ต COM ได้เพียง 1024 ไบต์เท่านั้น

### ตัวอย่างนิพจน์ 1

IO\_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 10)

จากตัวอย่างข้างบน เป็นการส่งข้อมูลจำนวน 10 ไบต์ที่จัดเก็บอยู่ในพื้นที่ LS1000 และพื้นที่ถัดไปจากพอร์ต COM

### ตัวอย่างนิพจน์ 2

IO\_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS0800])

จากตัวอย่างข้างบน เป็นการส่งข้อมูลที่จัดเก็บอยู่ในพื้นที่ LS1000 และพื้นที่ถัดไปจากพอร์ต COM จำนวนไบต์จะเหมือนกับที่เขียนใน LS0800

### ตัวอย่างนิพจน์ 3

IO\_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], [t:0010])

จากตัวอย่างข้างบน เป็นการส่งข้อมูลที่จัดเก็บอยู่ในพื้นที่ LS1000 และพื้นที่ถัดไปจากพอร์ต COM จำนวนไบต์จะเหมือนกับที่เขียนในตำแหน่งชั่วคราว [t:0010]

## โหมดการจัดเก็บข้อมูล

เมื่ออ่านข้อมูลจากตำแหน่งอุปกรณ์ขณะเรียกใช้ฟังก์ชันการทำงานของพอร์ต COM คุณสามารถระบุลำดับการจัดเก็บของข้อมูลได้อ่านได้

การตั้งค่าโหมดการจัดเก็บข้อมูลในพื้นที่ LS9130 สามารถเปลี่ยนลำดับการจัดเก็บได้

โดยสามารถเลือกโหมดได้จากตัวเลือกทั้งสี่ ได้แก่ โหมด 0, 1, 2 และ 3

### ◆ โหมด 0

ตัวอย่าง การใช้ฟังก์ชันการทำงานของพอร์ต COM อ่านสตริง “ABCDEFGH” จากตำแหน่งอุปกรณ์

[w:[#INTERNAL]LS9130] = 0

IO\_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 7)

- เมื่อตำแหน่งอุปกรณ์ยาว 16 บิต

|        |     |     |
|--------|-----|-----|
| LS0100 | 'A' | 'B' |
| LS0101 | 'C' | 'D' |
| LS0102 | 'E' | 'F' |
| LS0103 | 'G' | 0   |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

- เมื่อตำแหน่งอุปกรณ์ยาว 32 บิต

|        |     |     |     |     |
|--------|-----|-----|-----|-----|
| LS0100 | 'A' | 'B' | 'C' | 'D' |
| LS0101 | 'E' | 'F' | 'G' | 0   |
| LS0102 |     |     |     |     |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

### ◆ โหมด 1

ตัวอย่าง การใช้ฟังก์ชันการทำงานของพอร์ต COM อ่านสตริง “ABCDEFGH” จากตำแหน่งอุปกรณ์

[w:[#INTERNAL]LS9130] = 1

IO\_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 7)

- เมื่อตำแหน่งอุปกรณ์ยาว 16 บิต

|        |     |     |
|--------|-----|-----|
| LS0100 | 'B' | 'A' |
| LS0101 | 'D' | 'C' |
| LS0102 | 'F' | 'E' |
| LS0103 | 0   | 'G' |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

- เมื่อตำแหน่งอุปกรณ์ยาว 32 บิต

|        |     |     |     |     |
|--------|-----|-----|-----|-----|
| LS0100 | 'B' | 'A' | 'D' | 'C' |
| LS0101 | 'F' | 'E' | 0   | 'G' |
| LS0102 |     |     |     |     |

เขียน “0” เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

### ◆ โหมด 2

ตัวอย่าง การใช้ฟังก์ชันการทำงานของพอร์ต COM อ่านสตริง “ABCDEFGH” จากตำแหน่งอุปกรณ์

[w:[#INTERNAL]LS9130] = 2

IO\_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 7)

- เมื่อตำแหน่งอุปกรณ์ยาว 16 บิต

|        |     |     |
|--------|-----|-----|
| LS0100 | 'C' | 'D' |
| LS0101 | 'A' | 'B' |
| LS0102 | 'G' | 0   |
| LS0103 | 'E' | 'F' |

เขียน "0" เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

- เมื่อตำแหน่งอุปกรณ์ยาว 32 บิต

|        |     |     |     |     |
|--------|-----|-----|-----|-----|
| LS0100 | 'C' | 'D' | 'A' | 'B' |
| LS0101 | 0   | 'G' | 'E' | 'F' |
| LS0102 |     |     |     |     |

เขียน "0" เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

### ◆ โหมด 3

ตัวอย่าง การใช้ฟังก์ชันการทำงานของพอร์ต COM อ่านสตริง "ABCDEFGH" จากตำแหน่งอุปกรณ์

[w:[#INTERNAL]LS9130] = 3

IO\_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 7)

- เมื่อตำแหน่งอุปกรณ์ยาว 16 บิต

|        |     |     |
|--------|-----|-----|
| LS0100 | 'D' | 'C' |
| LS0101 | 'B' | 'A' |
| LS0102 | 0   | 'G' |
| LS0103 | 'F' | 'E' |

เขียน "0" เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

- เมื่อตำแหน่งอุปกรณ์ยาว 32 บิต

|        |     |     |     |     |
|--------|-----|-----|-----|-----|
| LS0100 | 'D' | 'C' | 'B' | 'A' |
| LS0101 | 0   | 'G' | 'F' | 'E' |
| LS0102 |     |     |     |     |

เขียน "0" เมื่อข้อมูลที่จะจัดเก็บมีจำนวนไบต์เป็นเลขคี่

### ข้อสำคัญ

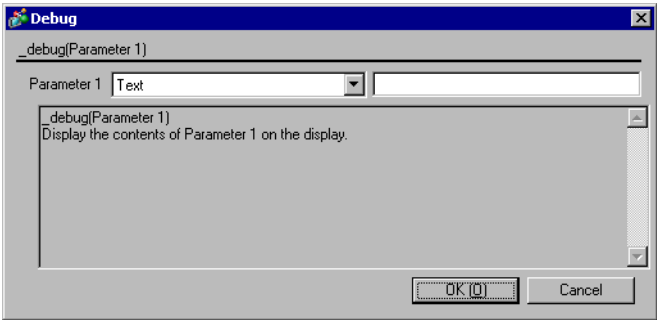
- โหมดการจัดเก็บข้อมูลไม่เหมือนกับโหมดข้อมูลสตริงในการตั้งค่าระบบ ความสัมพันธ์ระหว่างโหมดการจัดเก็บข้อมูลกับโหมดข้อมูลสตริงจะแสดงอยู่ในตารางด้านล่าง

| ลำดับการจัดเก็บของอุปกรณ์ข้อมูล | ลำดับการจัดเก็บไบต์ในเวิร์ดแบบ LH หรือ HL | ลำดับการจัดเก็บในดับเบิลเวิร์ดแบบ LH หรือ HL | โหมดการจัดเก็บข้อมูล D-Script | โหมดข้อมูลตัวอักษร |
|---------------------------------|-------------------------------------------|----------------------------------------------|-------------------------------|--------------------|
| จัดเก็บจากข้อมูลแรกสุด          | ลำดับ HL                                  | ลำดับ HL                                     | 0                             | 1                  |
|                                 | ลำดับ LH                                  |                                              | 1                             | 2                  |
|                                 | ลำดับ HL                                  | ลำดับ LH                                     | 2                             | 5                  |
|                                 | ลำดับ LH                                  |                                              | 3                             | 4                  |
| จัดเก็บจากข้อมูลท้ายสุด         | ลำดับ HL                                  | ลำดับ HL                                     | —                             | 3                  |
|                                 | ลำดับ LH                                  |                                              | —                             | 7                  |
|                                 | ลำดับ HL                                  | ลำดับ LH                                     | —                             | 8                  |
|                                 | ลำดับ LH                                  |                                              | —                             | 6                  |

## 21.7 อื่น ๆ

| Others                                                                            | ข้อมูลสรุปของฟังก์ชัน                                                                                                                        |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>ฟังก์ชัน Debug</p> <p>☞ “21.7.1 ฟังก์ชัน Debug” (หน้า 21-65)</p> <p>แสดงตำแหน่งหรือข้อความที่กำหนดบนหน้าจอเพื่อทำการตรวจแก้ข้อบกพร่อง</p> |

### 21.7.1 ฟังก์ชัน Debug

| รายการ     | คำอธิบาย                                                                                                                                                                                                              |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | แสดงตำแหน่งหรือข้อความที่กำหนดไว้บนหน้าจอเพื่อทำการตรวจแก้ข้อบกพร่อง หลังจากตรวจแก้ข้อบกพร่องเสร็จแล้ว ให้ยกเลิกการเลือกช่องทำเครื่องหมาย [Enable Debug Function] ของตัวแก้ไขสคริปต์ และฟังก์ชันนี้ก็จะหายไปจากหน้าจอ |
| รูปแบบ     | <p>_debug (Parameter 1)</p>  <p>Parameter 1: ข้อความ (อักขระแบบไบต์เดี่ยวสูงสุด 32 อักขระ)</p>                                     |

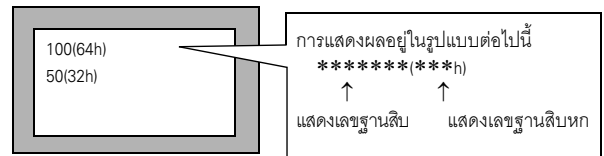
## ■ รายละเอียดของ Parameter 1

| Parameter 1                        | รูปแบบ           | คำอธิบาย                                                                            |
|------------------------------------|------------------|-------------------------------------------------------------------------------------|
| ข้อความ                            | _debug (“ABC”)   | แสดงข้อความในเครื่องหมาย “ ” ข้อความสามารถใช้อักขระแบบไบต์เดี่ยวได้สูงสุด 32 อักขระ |
| ตำแหน่งเวิร์ด หรือ ตำแหน่งชั่วคราว | _debug (w:D1000) | แสดงค่าของตำแหน่งเวิร์ดหรือตำแหน่งชั่วคราวที่กำหนดไว้                               |
| การขึ้นบรรทัดใหม่                  | _debug (_CRLF)   | เลื่อนเคอร์เซอร์ไปยังจุดเริ่มต้นของบรรทัดถัดไป                                      |
| การปิดแคร์                         | _debug (_CR)     | เลื่อนเคอร์เซอร์ไปยังจุดเริ่มต้นของบรรทัดเดียวกัน                                   |

### ตัวอย่างนิพจน์ 1

สคริปต์ต่อไปนี้จะแสดงค่าของตำแหน่งเวิร์ด

```
[w:[#INTERNAL]LS0100]=100
_debug ([w:[#INTERNAL]LS0100])
_debug (_CRLF)
[w:[#INTERNAL]LS0100]=50
_debug ([w:[#INTERNAL]LS0100])
```

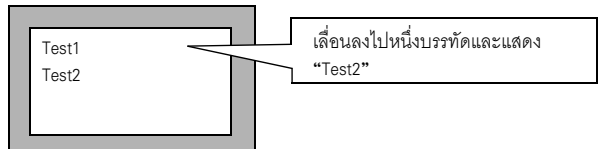


### ตัวอย่างนิพจน์ 2

สคริปต์ต่อไปนี้จะแสดงการขึ้นบรรทัดใหม่

และข้อความ

```
_debug (“Test1”)
_debug (_CRLF)
_debug (“Test2”)
```



## 21.8 นิพจน์คำอธิบาย

| นิพจน์คำอธิบาย                                                                                                                                                                                                            | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div>                     Description Expression<br/> <a href="#">if - endif</a><br/> <a href="#">if - else - endif</a><br/> <a href="#">loop - endloop</a><br/> <a href="#">break</a><br/> <a href="#">return</a> </div> | if - endif<br> “21.8.1 if - endif” (หน้า 21-67)<br>เมื่อเงื่อนไขภายในวงเล็บ “( )” ที่อยู่ตามหลัง “if” เป็นจริง ระบบจะดำเนินการกระบวนการที่อยู่หลังข้อความคำสั่ง “if ( )”                                                                                         |
|                                                                                                                                                                                                                           | if - else - endif<br> “21.8.2 if - else - endif” (หน้า 21-67)<br>เมื่อเงื่อนไขภายในวงเล็บ “( )” ที่อยู่ตามหลัง “if” เป็นจริง ระบบจะดำเนินการกระบวนการที่อยู่หลังข้อความคำสั่ง “if ( )” เมื่อเงื่อนไขเป็นเท็จ ระบบจะดำเนินการตามข้อความคำสั่งที่อยู่ถัดจาก “else” |
|                                                                                                                                                                                                                           | loop - endloop<br> “21.8.3 loop - endloop” (หน้า 21-68)<br>ประมวลผลแบบวนลูป (วนซ้ำ) เป็นจำนวนครั้งที่เท่ากับตัวเลขที่จัดเก็บอยู่ในตำแหน่งชั่วคราวซึ่งกำหนดอยู่ในวงเล็บ “( )” ที่อยู่หลัง “loop”                                                                  |
|                                                                                                                                                                                                                           | break<br> “21.8.4 break” (หน้า 21-70)<br>หยุดการวนลูปขณะกำลังรันสมการ loop ( )                                                                                                                                                                                   |
|                                                                                                                                                                                                                           | return<br> “21.8.5 return” (หน้า 21-70)<br>ดำเนินการอีกครั้งตั้งแต่เริ่มต้น<br>ตัวแปรนี้ใช้ได้เฉพาะใน Extended Script เท่านั้น                                                                                                                                   |

### 21.8.1 if - endif

เมื่อเงื่อนไขภายในวงเล็บ “( )” ที่อยู่ตามหลัง “if” เป็นจริง ระบบจะดำเนินการกระบวนการที่อยู่หลังข้อความคำสั่ง “if ( )”

**หมายเหตุ** • อักขระ “=” ในคำสั่ง Assign ใช้ไม่ได้ในนิพจน์เงื่อนไข

### 21.8.2 if - else - endif

เมื่อเงื่อนไขภายในวงเล็บ “( )” ที่อยู่ตามหลัง “if” เป็นจริง ระบบจะดำเนินการกระบวนการที่อยู่หลังข้อความคำสั่ง “if ( )” เมื่อเงื่อนไขเป็นเท็จ ระบบจะดำเนินการตามข้อความคำสั่งที่อยู่ถัดจาก “else”

**หมายเหตุ** • อักขระ “=” ในคำสั่ง Assign ใช้ไม่ได้ในนิพจน์เงื่อนไข

## 21.8.3 loop - endloop

ประมวลผลแบบวนลูป (วนซ้ำ) เป็นจำนวนครั้งเท่ากับตัวเลขที่จัดเก็บอยู่ในตำแหน่งชั่วคราวซึ่งกำหนดอยู่ในวงเล็บ “( )” ที่อยู่หลัง “loop”

## การวนลูปแบบไม่รู้จบ

เมื่อไม่ได้ป้อนข้อความคำสั่งในวงเล็บ ( ) ของข้อความคำสั่ง “loop ( )” การวนลูปจะถูกกำหนดเป็นการวนลูปแบบไม่รู้จบ

การวนลูปแบบไม่รู้จบสามารถใช้ได้เฉพาะใน Extended Script เท่านั้น

## ตัวอย่างนิพนธ์

```
loop ( )
{
    [w:[#INTERNAL]LS0100]=[w:[#INTERNAL]LS0100]+1
    if ( [w:[#INTERNAL]LS0100] >10)
    {
        break
    }
    endif
}
endloop
```

## หมายเหตุ

- รูปแบบของ loop ( ) จะเป็นดังต่อไปนี้

ตัวอย่าง

loop (จำนวนลูป) <= กำหนดตำแหน่งชั่วคราวที่จะกำหนดจำนวนครั้งในการวนลูปซ้ำ

```
{
    สมการการดำเนินการ
    break <= ระบุไว้เมื่อออกจากลูปหลังจากวนไปใดครั้งทาง (สามารถละได้)
} endloop <= ระบุไว้ที่ตอนท้ายของการวนลูป
```

- สามารถป้อนได้เฉพาะตำแหน่งเวิร์ดชั่วคราว (ในวงเล็บ) เท่านั้น (เช่น loop ([t:000]))
- ใช้ “loop ( )” กับสมการตรรกเกอร์ไม่ได้
- ค่าตำแหน่งเวิร์ดชั่วคราวที่ใช้กำหนด “การวนลูปแบบไม่รู้จบ” จะลดลงทุกครั้งที่ทำการวนลูป เมื่อค่าเปลี่ยนเป็น 0 การวนลูปจะสิ้นสุดลง หากมีการแก้ไขค่าตำแหน่งเวิร์ดชั่วคราวที่ใช้กำหนด “การวนลูปแบบไม่รู้จบ” การวนลูปจะดำเนินไปไม่รู้จบ นอกจากนี้ ตำแหน่งเวิร์ดชั่วคราวที่ใช้จะถูกกำหนดเป็นค่าที่ใช้เหมือนกันในทุกที่ ดังนั้น การใช้ตำแหน่งเวิร์ดชั่วคราวนี้พร้อมกันกับรายการอื่นอาจทำให้เกิดการวนลูปไม่รู้จบได้
- การแสดงพาร์ทบนหน้าจอ ฯลฯ จะไม่ถูกอัปเดต/รีเฟรชจนกว่าจะวนลูปเสร็จ

ต่อ

### หมายเหตุ

- loop ( ) ยังสามารถเขียนซ้อนกันได้ด้วย เมื่อเขียนการวนลูปซ้อนกัน loop ( ) ที่อยู่ด้านในสุด จะถูกข้ามไปโดยผ่านทางคำสั่ง “break”

```

loop ([t:0000])// ลูป 1
{
    loop ([t:0001])// ลูป 2
    {
        break// ออกจากลูป 2
    }endloop
}endloop
break// ออกจากลูป 1
}endloop

```

- หากการวนลูปเสร็จสิ้นลงโดยไม่ได้ใช้คำสั่ง escape ค่าตำแหน่งเวิร์ดชั่วคราวจะกลายเป็น 0
  - ช่วงค่าตำแหน่งเวิร์ดชั่วคราวที่สามารถใช้ได้จะแตกต่างกันไป โดยขึ้นอยู่กับรูปแบบของข้อมูล (Bin, BCD), ความยาวบิต และรหัส +/- ที่ใช้ หากกำหนดรหัส +/- และตำแหน่งเวิร์ดชั่วคราว เปลี่ยนเป็นค่าลบ ระบบจะพิจารณาเงื่อนไขที่ตอนเริ่มต้นของการวนลูป และการวนลูปจะหยุดลง
  - ห้ามใช้อุปกรณ์ PLC ในสูตรการวนลูป ให้ใช้ตำแหน่งจากอุปกรณ์ในพื้นที่สำหรับผู้ใช้งานพื้นที่ LS ภายในของ GP หรือตำแหน่งเวิร์ดชั่วคราวแทน
- ตัวอย่างเช่น คำอธิบายต่อไปนี้เขียนข้อมูลลงใน PLC หลายครั้งในช่วงเวลาสั้นๆ (100 ครั้ง ในตัวอย่างต่อไปนี้) ซึ่งอาจทำให้ระบบเกิดข้อผิดพลาดได้ เนื่องจากไม่สามารถประมวลผลการสื่อสาร (เวลาที่ต้องใช้ในการเขียนข้อมูลลงใน PLC) ด้วยความเร็วระดับนี้ได้
- ตัวอย่าง

```

[t:0000] = 100 // จำนวนลูป: 100
loop ([t:0000])
{
    [w:[PLC1]D0200] = [w:[#INTERNAL]LS0100] // เขียนลงใน D0200
    [w:[#INTERNAL]LS0100] = [w:[#INTERNAL]LS0100] + 1 // เพิ่มขึ้น LS0100
}endloop

```

โปรดเปลี่ยนให้เป็นดังนี้

```

[t:0000] = 100 // จำนวนลูป: 100
loop ([t:0000])
{
    [w:[#INTERNAL]LS0200] = [w:[#INTERNAL]LS0100] // เขียนลงใน D0200
    [w:[#INTERNAL]LS0100] = [w:[#INTERNAL]LS0100] + 1 // เพิ่มขึ้น LS0100
}endloop
[w:[PLC1]D0200]=[w:[#INTERNAL]LS0200] // เขียนข้อมูล LS0200 ลงใน D0200

```

- การใช้ “loop” หรือ “break” เป็นชื่อฟังก์ชันสำหรับฟังก์ชัน D-Script จะทำให้เกิดข้อผิดพลาด

## 21.8.4 break

หยุดการวนลูปขณะกำลังรันสมการ loop ( )

## หมายเหตุ

- สามารถใช้คำสั่ง “break” ได้เฉพาะในส่วน { } ของ loop ( ) เท่านั้น

## 21.8.5 return

เมื่อ “ฟังก์ชันที่กำหนดโดยผู้ใช้” มีคำสั่ง “return” อยู่ด้วย  
การประมวลผลฟังก์ชันสิ้นสุดลงและการควบคุมกลับไปให้ตัวเรียกของฟังก์ชัน

เมื่อการดำเนินงาน (ฟังก์ชันหลัก) มีคำสั่ง “return” อยู่ด้วย  
การประมวลผลฟังก์ชันหลักถูกยกเลิกครู่หนึ่ง และเริ่มต้นใหม่จากตอนเริ่มต้นของฟังก์ชันหลัก

## หมายเหตุ

- อักขระ “=” ในคำสั่ง Assign ใช้ไม่ได้ในนิพจน์เงื่อนไข

## ตัวอย่างนิพจน์

```
[w:[#INTERNAL]LS0100]=[w:[#INTERNAL]LS0200]>> 8) & 0xFF
if ([w:[#INTERNAL]LS0100]==0)           // เมื่อ LS0100 มีค่าเป็น “0” จะไม่มีการประมวลผล
                                         อีกต่อไป
{
    set([b:[#INTERNAL]LS005000])         // กำหนดตำแหน่งบิตสำหรับแสดงข้อผิดพลาด
    return                               // จบ
}
endif
```

## 21.9 การเปรียบเทียบ

| Comparison                                                                                                                                                                                                                                                                                                                                                                   | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Comparison<br><a href="#">Logical AND (AND)</a><br><a href="#">Logical OR (OR)</a><br><a href="#">Negation (not)</a><br><a href="#">less than (&lt;)</a><br><a href="#">less than or equal to (&lt;=)</a><br><a href="#">not equal to (&lt;&gt;)</a><br><a href="#">more than (&gt;)</a><br><a href="#">more than or equal to (&gt;=)</a><br><a href="#">Equivalent (==)</a> | Logical AND (AND)<br> “21.9.1 Logical AND (AND)” (หน้า 21-71)<br>N1 and N2: เป็นจริงหาก N1 และ N2 เปิดทั้งคู่                              |
|                                                                                                                                                                                                                                                                                                                                                                              | Logical OR (OR)<br> “21.9.2 Logical OR (OR)” (หน้า 21-71)<br>N1 and N2: เป็นจริงหาก N1 หรือ N2 ตัวใดตัวหนึ่งเปิด                           |
|                                                                                                                                                                                                                                                                                                                                                                              | Negation (not)<br> “21.9.3 Negation (not)” (หน้า 21-71)<br>notN1: กลายเป็น 0 หาก N1 เป็น 1 และกลายเป็น 1 หาก N1 เป็น 0                     |
|                                                                                                                                                                                                                                                                                                                                                                              | less than (<)<br> “21.9.4 less than (<)” (หน้า 21-72)<br>เป็นจริงหาก N1 น้อยกว่า N2 ( $N1 < N2$ )                                          |
|                                                                                                                                                                                                                                                                                                                                                                              | less than or equal to (<=)<br> “21.9.5 less than or equal to (<=)” (หน้า 21-72)<br>เป็นจริงหาก N1 น้อยกว่าหรือเท่ากับ N2 ( $N1 \leq N2$ )  |
|                                                                                                                                                                                                                                                                                                                                                                              | not equal to (<>)<br> “21.9.6 not equal to (<>)” (หน้า 21-72)<br>เป็นจริงหาก N1 ไม่เท่ากับ N2 ( $N1 \neq N2$ )                             |
|                                                                                                                                                                                                                                                                                                                                                                              | more than (>)<br> “21.9.7 more than (>)” (หน้า 21-72)<br>เป็นจริงหาก N1 มากกว่า N2 ( $N1 > N2$ )                                          |
|                                                                                                                                                                                                                                                                                                                                                                              | more than or equal to (>=)<br> “21.9.8 more than or equal to (>=)” (หน้า 21-72)<br>เป็นจริงหาก N1 มากกว่าหรือเท่ากับ N2 ( $N1 \geq N2$ ) |
|                                                                                                                                                                                                                                                                                                                                                                              | Equivalent (==)<br> “21.9.9 Equivalent (==)” (หน้า 21-72)<br>เป็นจริงหาก N1 เท่ากับ N2 ( $N1 = N2$ )                                     |

### 21.9.1 Logical AND (AND)

เชื่อมค่าทางขวากับทางซ้ายด้วยเงื่อนไข AND จะถือว่าค่า 0 (ศูนย์) คือปิด และค่าอื่นคือเปิด  
 N1 and N2: เป็นจริงหาก N1 และ N2 เปิดทั้งคู่ หากไม่ใช่ จะเป็นเท็จ

### 21.9.2 Logical OR (OR)

เชื่อมค่าทางขวาและทางซ้ายด้วยเงื่อนไข OR จะถือว่าค่า 0 (ศูนย์) คือปิด และค่าอื่นคือเปิด  
 N1 and N2: เป็นจริงหาก N1 หรือ N2 ตัวใดตัวหนึ่งเปิด หากไม่ใช่ จะเป็นเท็จ

### 21.9.3 Negation (not)

ใช้เงื่อนไข NOT กับค่าที่อยู่ทางขวา จะถือว่าค่า 0 (ศูนย์) คือ 1 และค่าอื่นคือ 0  
 notN1: กลายเป็น 0 หาก N1 เป็น 1 และกลายเป็น 1 หาก N1 เป็น 0

#### 21.9.4 less than (<)

เปรียบเทียบข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือข้อมูลในตำแหน่งเวิร์ดกับค่าคงที่หนึ่งค่า เป็นจริงหาก N1 น้อยกว่า N2 ( $N1 < N2$ )

#### 21.9.5 less than or equal to (<=)

เปรียบเทียบข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือข้อมูลในตำแหน่งเวิร์ดกับค่าคงที่หนึ่งค่า เป็นจริงหาก N1 น้อยกว่าหรือเท่ากับ N2 ( $N1 \leq N2$ )

#### 21.9.6 not equal to (<>)

เปรียบเทียบข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือข้อมูลในตำแหน่งเวิร์ดกับค่าคงที่หนึ่งค่า เป็นจริงหาก N1 ไม่เท่ากับ N2 ( $N1 \neq N2$ )

#### 21.9.7 more than (>)

เปรียบเทียบข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือข้อมูลในตำแหน่งเวิร์ดกับค่าคงที่หนึ่งค่า เป็นจริงหาก N1 มากกว่า N2 ( $N1 > N2$ )

#### 21.9.8 more than or equal to (>=)

เปรียบเทียบข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือข้อมูลในตำแหน่งเวิร์ดกับค่าคงที่หนึ่งค่า เป็นจริงหาก N1 มากกว่าหรือเท่ากับ N2 ( $N1 \geq N2$ )

#### 21.9.9 Equivalent (==)

เปรียบเทียบข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือข้อมูลในตำแหน่งเวิร์ดกับค่าคงที่หนึ่งค่า เป็นจริงหาก N1 เท่ากับ N2 ( $N1 = N2$ )

| คำสั่ง              |     | ตัวอย่าง                       |
|---------------------|-----|--------------------------------|
| การเชื่อม           | and | if ((การทำงาน) and (การทำงาน)) |
| การเลือก            | or  | if ((การทำงาน) or (การทำงาน))  |
| นิเสธ               | not | if (not (การทำงาน))            |
| น้อยกว่า            | <   | (ค่า 1) < (ค่า 2)              |
| น้อยกว่าหรือเท่ากับ | <=  | (ค่า 1) <= (ค่า 2)             |
| ไม่เท่ากับ          | <>  | (ค่า 1) <> (ค่า 2)             |
| มากกว่า             | >   | (ค่า 1) > (ค่า 2)              |
| มากกว่าหรือเท่ากับ  | >=  | (ค่า 1) >= (ค่า 2)             |
| เท่ากับ             | ==  | (ค่า 1) == (ค่า 2)             |

## 21.10 ตัวดำเนินการ

| Operator                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Operator</b><br><a href="#">Addition (+)</a><br><a href="#">Subtraction (-)</a><br><a href="#">Margin (%)</a><br><a href="#">Multiplication (*)</a><br><a href="#">Division (/)</a><br><a href="#">Assignment (=)</a><br><a href="#">Left Shift (&lt;&lt;)</a><br><a href="#">Right Shift (&gt;&gt;)</a><br><a href="#">Bit Operator Logical AND (&amp;)</a><br><a href="#">Bit Operator Logical OR ( )</a><br><a href="#">Bit Operator Exclusive OR (^)</a><br><a href="#">Bit Operator 1's Complement (~)</a> | Addition (+)<br> “21.10.1 Addition (+)” (หน้า 21-74)<br>เพิ่มข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือเพิ่มข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า                                                      |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Subtraction (-)<br> “21.10.2 Subtraction (-)” (หน้า 21-74)<br>ลบข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือลบข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า                                                      |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Margin (%)<br> “21.10.3 Margin (%)” (หน้า 21-74)<br>ตรวจสอบเศษที่เหลือจากการหารข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือเศษที่เหลือจากการหารข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า                     |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Multiplication (*)<br> “21.10.4 Multiplication (*)” (หน้า 21-74)<br>คูณข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือคูณข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า                                              |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Division (/)<br> “21.10.5 Division (/)” (หน้า 21-74)<br>หารข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือหารข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า                                                          |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Assignment (=)<br> “21.10.6 Assignment (=)” (หน้า 21-74)<br>กำหนดค่าฝั่งขวาในค่าฝั่งซ้าย                                                                                                                      |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Left Shift (<<)<br> “21.10.7 Left Shift (<<)” (หน้า 21-75)<br>เลื่อนข้อมูลฝั่งซ้ายไปทางซ้ายตามจำนวนที่อยู่ในฝั่งขวา                                                                                           |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Right Shift (>>)<br> “21.10.8 Right Shift (>>)” (หน้า 21-75)<br>เลื่อนข้อมูลฝั่งซ้ายไปทางขวาตามจำนวนที่อยู่ในฝั่งขวา                                                                                          |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Bit Operator Logical AND (&)<br> “21.10.9 Bit Operator Logical AND (&)” (หน้า 21-75)<br>ใช้ตัวดำเนินการ logical AND กับข้อมูลระหว่างอุปกรณ์ชนิดเวิร์ดสองอุปกรณ์ หรือระหว่างข้อมูลอุปกรณ์ชนิดเวิร์ดและค่าคงที่ |

ต่อ

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Operator</b><br><a href="#">Addition (+)</a><br><a href="#">Subtraction (-)</a><br><a href="#">Margin (%)</a><br><a href="#">Multiplication (*)</a><br><a href="#">Division (/)</a><br><a href="#">Assignment (=)</a><br><a href="#">Left Shift (&lt;&lt;)</a><br><a href="#">Right Shift (&gt;&gt;)</a><br><a href="#">Bit Operator Logical AND (&amp;)</a><br><a href="#">Bit Operator Logical OR ( )</a><br><a href="#">Bit Operator Exclusive OR (^)</a><br><a href="#">Bit Operator 1's Complement (~)</a> | Bit Operator Logical OR ( )<br> “21.10.10 Bit Operator Logical OR ( )” (หน้า 21-75)<br>ใช้ตัวดำเนินการ logical OR กับข้อมูลระหว่างอุปกรณ์ชนิดเวิร์ดสองอุปกรณ์ หรือระหว่างข้อมูลอุปกรณ์ชนิดเวิร์ดและค่าคงที่       |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Bit Operator Exclusive OR (^)<br> “21.10.11 Bit Operator Exclusive OR (^)” (หน้า 21-75)<br>ใช้ตัวดำเนินการ exclusive OR กับข้อมูลระหว่างอุปกรณ์ชนิดเวิร์ดสองอุปกรณ์ หรือระหว่างข้อมูลอุปกรณ์ชนิดเวิร์ดและค่าคงที่ |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Bit Operator 1's Complement (~)<br> “21.10.12 Bit Operator 1's Complement (~)” (หน้า 21-75)<br>สลับบิตเป็นตรงกันข้าม                                                                                              |

### 21.10.1 Addition (+)

เพิ่มข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือเพิ่มข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า Overflowing Digit ใดๆ ที่เป็นผลจากการคำนวณจะถูกปิดเศษ

### 21.10.2 Subtraction (-)

ลบข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือลบข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า Overflowing Digit ใดๆ ที่เป็นผลจากการคำนวณจะถูกปิดเศษ

### 21.10.3 Margin (%)

ตรวจหาเศษที่เหลือจากการหารข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือเศษที่เหลือจากการหารข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า ผลการคำนวณอาจขึ้นอยู่กับสัญลักษณ์ทางฝั่งซ้ายและฝั่งขวา

### 21.10.4 Multiplication (\*)

คูณข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือคูณข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า Overflowing Digit ใดๆ ที่เป็นผลจากการคำนวณจะถูกปิดเศษ

### 21.10.5 Division (/)

หารข้อมูลในตำแหน่งเวิร์ดสองตำแหน่ง หรือหารข้อมูลในตำแหน่งเวิร์ดหนึ่งตำแหน่งและค่าคงที่หนึ่งค่า ตำแหน่งทศนิยมที่เป็นผลจากการคำนวณจะถูกปิดเศษ Overflowing Digit ใดๆ ที่เป็นผลจากการคำนวณจะถูกปิดเศษ

### 21.10.6 Assignment (=)

กำหนดค่าฝั่งขวาในค่าฝั่งซ้าย ค่าฝั่งซ้ายสามารถระบุได้เฉพาะตำแหน่งอุปกรณ์เท่านั้น ค่าฝั่งขวาสามารถเป็นได้ทั้งตำแหน่งอุปกรณ์และค่าคงที่ Overflowing Digit ใดๆ ที่เป็นผลจากการคำนวณจะถูกปิดเศษ



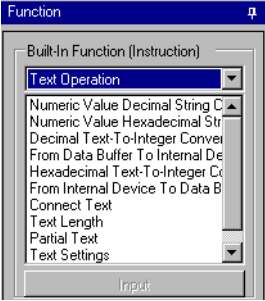
## ลำดับความสำคัญและการเชื่อมโยง

ตารางต่อไปนี้แสดงลำดับความสำคัญของเงื่อนไขการทริกเกอร์ หากตัวดำเนินการสองตัวขึ้นไปมีลำดับความสำคัญเท่ากัน ให้ไปตามทิศทางที่แสดงไว้ในตารางเชื่อมโยง

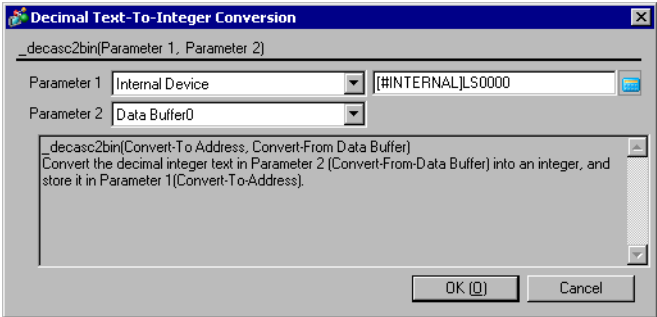
| ลำดับความสำคัญ | ตัวดำเนินการ | การเชื่อมโยง |
|----------------|--------------|--------------|
| สูง            | ( )          | →            |
|                | not ~        | ←            |
|                | * / %        | →            |
|                | + -          | →            |
|                | << >>        | →            |
|                | < <= > >=    | →            |
|                | == <>        | →            |
|                | & ^          | →            |
|                | and or       | →            |
|                | =            | ←            |
| ต่ำ            |              |              |

## 21.11 การทำงานของข้อความ

ฟังก์ชันการทำงานของข้อความสามารถใช้ได้เฉพาะใน Extended Script เท่านั้น

| Text Operation                                                                     | ข้อมูลสรุปของฟังก์ชัน                                                                                                                                                              |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Decimal Text-To-Integer Conversion<br>“21.11.1 Decimal Text-To-Integer Conversion” (หน้า 21-78)<br>ฟังก์ชันนี้ใช้แปลงข้อความเลขฐานสิบให้เป็นจำนวนเต็ม                              |
|                                                                                    | Hexadecimal Text-To-Integer Conversion<br>“21.11.2 Hexadecimal Text-To-Integer Conversion” (หน้า 21-80)<br>ฟังก์ชันนี้ใช้แปลงข้อความเลขฐานสิบหกให้เป็นจำนวนเต็ม                    |
|                                                                                    | From Internal Device To Data Buffer<br>“21.11.3 From Internal Device To Data Buffer” (หน้า 21-82)<br>คัดลอกข้อมูลของสตริงที่จัดเก็บในอุปกรณ์ภายในไปยังบัฟเฟอร์ข้อมูล               |
|                                                                                    | From Data Buffer to Internal Device<br>“21.11.4 From Data Buffer to Internal Device” (หน้า 21-84)<br>คัดลอกข้อมูลของสตริงที่จัดเก็บในบัฟเฟอร์ข้อมูลไปยังอุปกรณ์ภายใน               |
|                                                                                    | Status<br>“21.11.5 สถานะข้อผิดพลาดของฟังก์ชันการทำงานของข้อความ” (หน้า 21-86)<br>จัดเก็บข้อผิดพลาดต่าง ๆ ที่เกิดขึ้น                                                               |
|                                                                                    | Numeric Value Decimal String Conversion<br>“21.11.6 Numeric Value Decimal String Conversion” (หน้า 21-88)<br>ใช้ฟังก์ชันนี้เพื่อแปลงจำนวนเต็มให้เป็นสตริงเลขฐานสิบ                 |
|                                                                                    | Numeric Value Hexadecimal String Conversion<br>“21.11.7 Numeric Value Hexadecimal String Conversion” (หน้า 21-89)<br>ใช้ฟังก์ชันนี้เพื่อแปลงข้อมูลเลขฐานสองให้เป็นสตริงเลขฐานสิบหก |
|                                                                                    | Partial Text<br>“21.11.8 Partial Text” (หน้า 21-90)<br>ค้นข้อมูลตามความยาวของสตริงจากออฟเซตที่ระบุไว้ของสตริง แล้วจัดเก็บข้อมูลไว้ในบัฟเฟอร์ข้อมูลอื่น                             |
|                                                                                    | Text Settings<br>“21.11.9 Text Settings” (หน้า 21-91)<br>จัดเก็บสตริงแบบตายตัวไว้ในบัฟเฟอร์ข้อมูล                                                                                  |
|                                                                                    | Text Length<br>“21.11.10 Text Length” (หน้า 21-92)<br>รับความยาวของสตริงที่จัดเก็บไว้                                                                                              |
|                                                                                    | Connect Text<br>“21.11.11 Connect Text” (หน้า 21-93)<br>เชื่อมสตริงอักขระหรือรหัสอักขระด้วยบัฟเฟอร์ข้อความ                                                                         |

### 21.11.1 Decimal Text-To-Integer Conversion

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                      |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ฟังก์ชันนี้ใช้แปลงสตริงเลขฐานสิบให้เป็นจำนวนเต็ม โดยแปลงข้อความจำนวนเต็มเลขฐานสิบใน Parameter 2 (บัพเฟอร์ข้อมูลที่ถูกแปลง) ให้เป็นจำนวนเต็ม และจัดเก็บไว้ใน Parameter 1 (ตำแหน่งที่แปลงแล้ว)                                                  |
| รูปแบบ     | <p>_decasc2bin ([ตำแหน่งที่แปลงแล้ว], [บัพเฟอร์ข้อมูลที่ถูกแปลง])</p>  <p>Parameter 1: อุปกรณ์ภายใน, ตำแหน่งชั่วคราว</p> <p>Parameter 2: บัพเฟอร์ข้อมูล</p> |

#### ตัวอย่างนิพจน์ 1 (เมื่อข้อมูลยาว 16 บิต)

\_decasc2bin ([w:[#INTERNAL]LS0100], databuf0)

รายละเอียดของ “databuf0” มีดังนี้

|             | 8 บิต |          |
|-------------|-------|----------|
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

ข้อมูลข้างบนจะถูกแปลงดังต่อไปนี้

|        | 16 บิต |
|--------|--------|
| LS0100 | 1234   |

## ตัวอย่างนิพจน์ 2 (เมื่อข้อมูลยาว 32 บิต)

`_decasc2bin ([w:[#INTERNAL]LS0100], databuf0)`

รายละเอียดของ “databuf0” มีดังนี้

|             | 8 บิต |          |
|-------------|-------|----------|
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 35h   | '5'      |
| databuf0[5] | 36h   | '6'      |
| databuf0[6] | 37h   | '7'      |
| databuf0[7] | 38h   | '8'      |
| databuf0[8] | 00h   | ค่าศูนย์ |

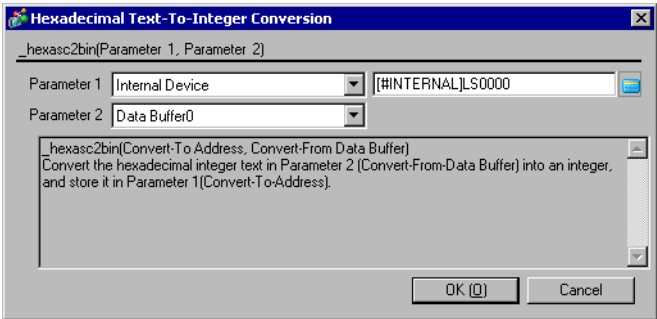
ข้อมูลข้างบนจะถูกแปลงดังต่อไปนี้

|        | 32 บิต   |
|--------|----------|
| LS0100 | 12345678 |
| LS0102 |          |

### ข้อสำคัญ

- เมื่อความยาวบิตที่แปลงแล้วมากกว่าความยาวบิตของ D-Script Editor จะเกิดข้อผิดพลาดขึ้น ตัวอย่าง เมื่อสคริปต์มีความยาวบิตเท่ากับ 16 บิต  
`_strset (databuf0, “123456”) // เมื่อป้อนสตริงเลขฐานสิบ 6 ตัวโดยไม่ตั้งใจ`  
`_decasc2bin ([w:[#INTERNAL]LS0100], databuf0)`  
 เมื่อเรียกใช้นิพจน์ข้างบน ข้อผิดพลาดหมายเลข 2 (ข้อผิดพลาดในการแปลงสตริง) ของสถานะข้อผิดพลาดของสตริง [e: STR\_ERR\_STAT] จะถูกทริกเกอร์ อย่างไรก็ตาม บิตจะกลับไปตอนเริ่มต้นของฟังก์ชันหลักเมื่อเกิดข้อผิดพลาด คุณจึงไม่สามารถอ้างอิงถึงฟังก์ชันอื่นโดยตรงได้หลังจากเรียกใช้ `_decasc2bin` แล้ว (หากมีคำสั่งเข้ามาระหว่างฟังก์ชันกำลังทำงาน ระบบจะกลับไปบรรทัดที่เรียกฟังก์ชันนั้นขึ้นมา)
- การแปลงสตริงข้อมูลที่มีอักขระอื่นที่ไม่ใช่ “0” ถึง “9” อยู่จะเกิดข้อผิดพลาดขึ้น ตัวอย่าง เมื่อสคริปต์มีความยาวบิตเท่ากับ 16 บิต  
`_strset (databuf0, “12AB”) // เมื่อป้อนสตริงที่ไม่ใช่เลขฐานสิบโดยไม่ตั้งใจ`  
`_decasc2bin ([w:[#INTERNAL]LS0100], databuf0)`  
 เมื่อเรียกใช้นิพจน์ข้างบน ข้อผิดพลาดหมายเลข 2 (ข้อผิดพลาดในการแปลงสตริง) ของสถานะข้อผิดพลาดของสตริง [e: STR\_ERR\_STAT] จะถูกทริกเกอร์ อย่างไรก็ตาม บิตจะกลับไปตอนเริ่มต้นของฟังก์ชันหลักเมื่อเกิดข้อผิดพลาด คุณจึงไม่สามารถอ้างอิงถึงฟังก์ชันอื่นโดยตรงได้หลังจากเรียกใช้ `_decasc2bin` แล้ว (หากมีคำสั่งเข้ามาระหว่างฟังก์ชันกำลังทำงาน ระบบจะกลับไปบรรทัดที่เรียกฟังก์ชันนั้นขึ้นมา)
- การประมวลผลจะสิ้นสุดลงเมื่อเกิดข้อผิดพลาดและกลับไปตอนเริ่มต้นของฟังก์ชันหลัก (หากมีคำสั่งเข้ามาระหว่างฟังก์ชันกำลังทำงาน ระบบจะกลับไปบรรทัดที่เรียกฟังก์ชันนั้นขึ้นมา)

## 21.11.2 Hexadecimal Text-To-Integer Conversion

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                      |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ฟังก์ชันนี้จะแปลงสตริงเลขฐานสิบหกให้เป็นข้อมูลเลขฐานสอง โดยแปลงข้อความจำนวนเต็มเลขฐานสิบหกใน Parameter 2 (บัพเฟอร์ข้อมูลที่ถูกแปลง) ให้เป็นจำนวนเต็ม และจัดเก็บไว้ใน Parameter 1 (ตำแหน่งที่แปลงแล้ว)                                         |
| รูปแบบ     | <p>_hexasc2bin ([ตำแหน่งที่แปลงแล้ว], [บัพเฟอร์ข้อมูลที่ถูกแปลง])</p>  <p>Parameter 1: อุปกรณ์ภายใน, ตำแหน่งชั่วคราว</p> <p>Parameter 2: บัพเฟอร์ข้อมูล</p> |

### ตัวอย่างนิพจน์ 1 (เมื่อข้อมูลยาว 16 บิต)

\_hexasc2bin ([w:[#INTERNAL]LS0100], databuf0)

รายละเอียดของ “databuf0” มีดังนี้

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

ข้อมูลข้างบนจะถูกแปลงดังต่อไปนี้

|        |        |
|--------|--------|
|        | 16 บิต |
| LS0100 | 1234h  |

## ตัวอย่างนิพจน์ 2 (เมื่อข้อมูลยาว 32 บิต)

\_hexasc2bin ([w:[#INTERNAL]LS0100], databuf0)

รายละเอียดของ “databuf0” มีดังนี้

| 8 บิต       |     |          |
|-------------|-----|----------|
| databuf0[0] | 31h | '1'      |
| databuf0[1] | 32h | '2'      |
| databuf0[2] | 33h | '3'      |
| databuf0[3] | 34h | '4'      |
| databuf0[4] | 35h | '5'      |
| databuf0[5] | 36h | '6'      |
| databuf0[6] | 37h | '7'      |
| databuf0[7] | 38h | '8'      |
| databuf0[8] | 00h | ค่าศูนย์ |

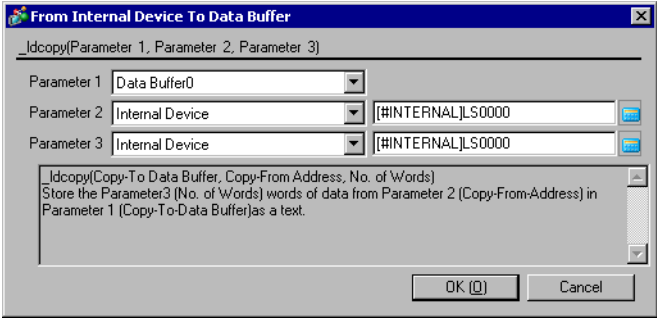
ข้อมูลข้างบนจะถูกแปลงดังต่อไปนี้

| 32 บิต |           |
|--------|-----------|
| LS0100 | 12345678h |
| LS0102 |           |

### ข้อสำคัญ

- เมื่อสตริงที่แปลงมีขนาดใหญ่กว่า 16 บิตหรือ 32 บิต จะเกิดข้อผิดพลาดขึ้น  
ตัวอย่าง เมื่อสคริปต์มีความยาวบิตเท่ากับ 16 บิต  
\_strset (databuf0, “123456”)  
\_hexasc2bin ([w:[#INTERNAL]LS0100], databuf0)  
เมื่อเรียกใช้นิพจน์ข้างบน ข้อผิดพลาดหมายเลข 2 (ข้อผิดพลาดในการแปลงสตริง) ของสถานะข้อผิดพลาดของสตริง [e: STR\_ERR\_STAT] จะถูกทริกเกอร์
- การแปลงสตริงข้อมูลที่มีอักขระอื่นที่ไม่ใช่ “0” ถึง “9”, “A” ถึง “F” หรือ “a” ถึง “f” จะเกิดข้อผิดพลาดขึ้น  
ตัวอย่าง เมื่อสคริปต์มีความยาวบิตเท่ากับ 16 บิต  
\_strset (databuf0, “123G”)  
\_hexasc2bin ([w:[#INTERNAL]LS0100], databuf0)  
เมื่อเรียกใช้นิพจน์ข้างบน ข้อผิดพลาดหมายเลข 2 (ข้อผิดพลาดในการแปลงสตริง) ของสถานะข้อผิดพลาดของสตริง [e: STR\_ERR\_STAT] จะถูกทริกเกอร์
- การประมวลผลจะสิ้นสุดลงเมื่อเกิดข้อผิดพลาดและกลับไปขั้นตอนเริ่มต้นของฟังก์ชันหลัก (หากมีคำสั่งเข้ามาระหว่างฟังก์ชันกำลังทำงาน ระบบจะกลับไปบรรทัดที่เรียกฟังก์ชันนั้นขึ้นมา)

### 21.11.3 From Internal Device To Data Buffer

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                        |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ข้อมูลของสตริงที่จัดเก็บอยู่ในพื้นที่ LS จะถูกคัดลอกไปยังบัฟเฟอร์ข้อมูลตามจำนวนสตริงในการถ่ายโอนแบบไบนารีต่อไบนารี ฟังก์ชันนี้จะจัดเก็บข้อมูลตามจำนวนเวิร์ดของ Parameter 3 (จำนวนเวิร์ด) จาก Parameter 2 (ตำแหน่งที่ถูกคัดลอก) ใน Parameter 1 (บัฟเฟอร์ข้อมูลที่คัดลอกแล้ว) ให้เป็นข้อความ                                                                                      |
| รูปแบบ     | <p>_Idcopy (บัฟเฟอร์ข้อมูลปลายทางการคัดลอก, [ตำแหน่งที่ถูกคัดลอก], จำนวนเวิร์ด)</p>  <p>Parameter 1: บัฟเฟอร์ข้อมูล<br/>         Parameter 2: อุปกรณ์ภายใน<br/>         Parameter 3: ค่าจำนวนเต็ม, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 3 คือตั้งแต่ 1 ถึง 1,024)</p> |

#### ตัวอย่างนิพจน์ 1

\_Idcopy (databuf0, [w:[#INTERNAL]LS0100], 4)

|        |        |
|--------|--------|
|        | 16 บิต |
| LS0100 | 31h    |
| LS0101 | 32h    |
| LS0102 | 33h    |
| LS0103 | 34h    |

ข้อมูลในพื้นที่ LS0100 ถึง LS0103 จะถูกเขียนลงในบัฟเฟอร์ข้อมูล 4 ไบต์ตามลำดับโดยเริ่มจาก “databuf0” พื้นที่ LS ถูกอ่านทีละไบต์ (บิตต่ำที่สุด)

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

**ข้อสำคัญ**

- ระบบจะอ่านไบต์ล่าง 1 ไบต์ของพื้นที่ LS และเขียนข้อมูลตามจำนวนที่ระบุลงในบัฟเฟอร์ข้อมูล
- ค่าตัวเลขสูงสุดที่สามารถกำหนดให้สำหรับ Parameter 3 คือ 1,024 เมื่อมีการตั้งค่าที่เกินกว่าที่กำหนดไว้ ข้อผิดพลาดหมายเลข 1 (สตริงโอเวอร์โฟลว์) ของสถานะข้อผิดพลาดของสตริง [e: STR\_ERR\_STAT] จะถูกทริกเกอร์
- แม้จะจัดเก็บข้อมูลไว้ในไบต์สูงสุดในพื้นที่ LS แต่ระบบจะอ่านเฉพาะข้อมูลในไบต์ล่าง 1 ไบต์เท่านั้น
- การประมวลผลจะสิ้นสุดลงเมื่อเกิดข้อผิดพลาดและกลับไปที่ยอนเริ่มต้นของฟังก์ชันหลัก (หากมีคำสั่งเข้ามาระหว่างฟังก์ชันกำลังทำงาน ระบบจะกลับไปที่ยอนที่เรียกฟังก์ชันนั้นขึ้นมา)

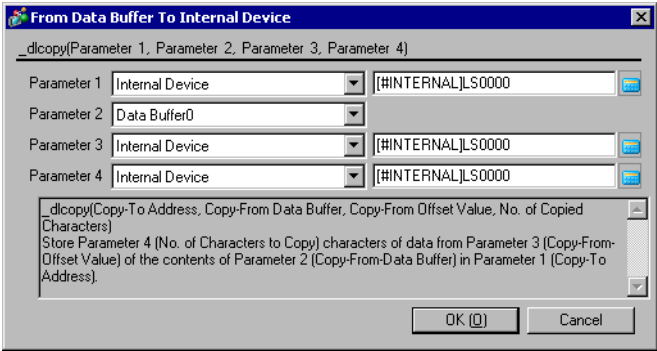
\_ldcopy (databuf0, [w:[#INTERNAL]LS0100], 4)

|        | 16 บิต |
|--------|--------|
| LS0100 | 3132h  |
| LS0101 | 3334h  |
| LS0102 | 3536h  |
| LS0103 | 3738h  |

- เมื่อจัดเก็บข้อมูลไว้ตามที่แสดงข้างบนแล้ว ระบบจะอ่านข้อมูลของไบต์ล่าง 1 ไบต์และเขียนลงในบัฟเฟอร์ข้อมูล

|             | 8 บิต |          |
|-------------|-------|----------|
| databuf0[0] | 32h   | '2'      |
| databuf0[1] | 34h   | '4'      |
| databuf0[2] | 36h   | '6'      |
| databuf0[3] | 38h   | '8'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

## 21.11.4 From Data Buffer to Internal Device

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ฟังก์ชันนี้จะคัดลอกข้อมูลสตริงแต่ละไบต์ที่เก็บไว้ในออฟเซตของบัฟเฟอร์ข้อมูลไปยังพื้นที่ LS ตามจำนวนสตริง<br>ฟังก์ชันจะจัดเก็บข้อมูลตามจำนวนอักขระของ Parameter 4 (จำนวนอักขระที่จะคัดลอก)<br>จาก Parameter 3 (ค่าออฟเซตที่ถูกคัดลอก) ของรายละเอียดของ Parameter 2<br>(บัฟเฟอร์ข้อมูลที่ถูกคัดลอก) ไว้ใน Parameter 1 (ตำแหน่งปลายทางการคัดลอก)                                                                                                                                                                             |
| รูปแบบ     | <p>_dlcopy ([ตำแหน่งปลายทางการคัดลอก], บัฟเฟอร์ข้อมูลที่ถูกคัดลอก, ค่าออฟเซตที่ถูกคัดลอก, จำนวนอักขระที่จะคัดลอก)</p>  <p>Parameter 1: อุปกรณ์ภายใน</p> <p>Parameter 2: บัฟเฟอร์ข้อมูล</p> <p>Parameter 3: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 3 คือตั้งแต่ 1 ถึง 1,024)</p> <p>Parameter 4: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 4 คือตั้งแต่ 1 ถึง 1,024)</p> |

### ตัวอย่างนิพจน์ 1

\_dlcopy ([w:[#INTERNAL]LS0100], databuf0, 2, 4)

|             | 8 บิต |     |
|-------------|-------|-----|
| databuf0[0] | 31h   | '1' |
| databuf0[1] | 32h   | '2' |
| databuf0[2] | 33h   | '3' |
| databuf0[3] | 34h   | '4' |
| databuf0[4] | 35h   | '5' |
| databuf0[5] | 36h   | '6' |
| databuf0[6] | 37h   | '7' |
| databuf0[7] | 38h   | '8' |

ระบบจะเขียนข้อมูลจำนวน 4 ไบต์ที่ค้นจาก “offset 2” ของ “databuf0” ลงในพื้นที่ LS0100 ถึง LS0103 โดยเขียนข้อมูลลงในพื้นที่ LS ทีละ 1 ไบต์

|        | 16 บิต |
|--------|--------|
| LS0100 | 33h    |
| LS0101 | 34h    |
| LS0102 | 35h    |
| LS0103 | 36h    |

**ข้อสำคัญ**

- ระบบจะอ่านข้อมูล 1 ไบต์จากบัฟเฟอร์ข้อมูลและเขียนลงในพื้นที่ LS ซึ่งหมายความว่าจะใช้เฉพาะ 8 บิตต่ำสุด (1 ไบต์) ของพื้นที่ LS เท่านั้น ข้อมูล 8 บิตสูงสุด (1 ไบต์) จะถูกล้างด้วยค่า “0”
  - เมื่อค่าที่ระบุ [ค่าออฟเซตต้นทาง + จำนวนอักขระที่จะคัดลอก] มากกว่าขนาดบัฟเฟอร์ข้อมูล ระบบจะแสดงข้อผิดพลาดหมายเลข 3 (ข้อผิดพลาดในการดึงสตริง) ของสถานะข้อผิดพลาดของสตริง [e: STR\_ERR\_STAT]
  - การประมวลผลจะสิ้นสุดลงเมื่อเกิดข้อผิดพลาดและกลับไปที่ยอนเริ่มต้นของฟังก์ชันหลัก (หากมีคำสั่งเข้ามาระหว่างฟังก์ชันกำลังทำงาน ระบบจะกลับไปที่ยอนที่เรียกฟังก์ชันนั้นขึ้นมา)
-

### 21.11.5 สถานะข้อผิดพลาดของฟังก์ชันการทำงานของข้อความ

เมื่อเกิดข้อผิดพลาดระหว่างเรียกใช้ฟังก์ชันการทำงานของข้อความ ระบบจะกำหนดข้อผิดพลาดให้กับสถานะข้อผิดพลาดของฟังก์ชันการทำงานของข้อความ [e: STR\_ERR\_STAT] หาก [e: STR\_ERR\_STAT] มีค่า “0” แสดงว่าฟังก์ชันทำงานได้เป็นปกติ แต่หากมีค่าอื่นที่ไม่ใช่ “0” แสดงว่ามีข้อผิดพลาดเกิดขึ้น ข้อผิดพลาดล่าสุดจะเก็บไว้ในสถานะข้อผิดพลาดของฟังก์ชันการทำงานของข้อความ [e: STR\_ERR\_STAT] คุณสามารถตั้งค่าสถานะข้อผิดพลาดของฟังก์ชันการทำงานของข้อความโดยใช้ฟังก์ชัน [SIO Port Operation/Label Settings] ที่อยู่ใต้เมนูกล่องเครื่องมือ D-Script ตารางต่อไปนี้แสดงรายการข้อผิดพลาดของฟังก์ชันการทำงานของข้อความ

| หมายเลขข้อผิดพลาด | ชื่อข้อผิดพลาด           | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                 | ปกติ                     | ไม่มีข้อผิดพลาด                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 1                 | ข้อความโอเวอร์โฟลว์      | มีการป้อนสตริงขนาดตั้งแต่ 256 ไบต์ขึ้นไปในอาร์กิวเมนต์ของฟังก์ชันต่อไปนี้โดยตรง ได้แก่ _strset ( ), _strlen ( ), _strcat ( ), _strmid ( ) และ IO_READ_WAIT ( )<br>หรือสร้างสตริงที่เกินกว่าขนาดบัพเฟอร์ข้อมูลในระหว่างเรียกใช้ฟังก์ชัน _strcat ( ) หรือ _ldcopy ( )<br>ตัวอย่าง<br>_strcat (databuf0, databuf1)<br>ระบบจะเรียกใช้ฟังก์ชันข้างบนเมื่อจัดเก็บสตริงขนาด 1,020 ไบต์ใน databuf0 และจัดเก็บสตริงขนาด 60 ไบต์ใน databuf1 (สตริงที่เกินกว่าบัพเฟอร์ข้อมูลซึ่งมีขนาดเท่ากับ 1,024 ไบต์จะทำให้เกิดสถานะข้อผิดพลาด) |
| 2                 | ข้อผิดพลาดในการแปลงสตริง | มีการป้อนรหัสอักขระที่ไม่ได้ไว้ในฟังก์ชัน _hexasc2bin ( ) หรือ _decasc2bin ( )<br>ตัวอย่าง<br>มีการใส่รหัสอักขระอื่นที่ไม่ใช่ “0” ถึง “9”, “A” ถึง “F” หรือ “a” ถึง “f” ไว้ในอาร์กิวเมนต์ที่สองของ _hexasc2bin ( )                                                                                                                                                                                                                                                                                                       |
| 3                 | ข้อผิดพลาดในการค้นสตริง  | มีการค้นหาสตริงอักขระที่ยาวกว่าที่ระบุไว้ด้วยฟังก์ชัน “_strmid ( )” หรือกำหนดค่าออฟเซตมากกว่าสตริงที่ระบุไว้<br>ตัวอย่าง<br>_strmid (databuf0, “12345678”, 2, 8)<br>ค้นหาสตริง 8 อักขระจากออฟเซต 2                                                                                                                                                                                                                                                                                                                       |

สถานะข้อผิดพลาดในการควบคุมสตริงใช้กับ D-Scripts และ Global D-Scripts ไม่ได้ หากมีการอ่านสถานะนั้นโดยไม่ตั้งใจ ระบบจะโหลดค่า “0” และจัดเก็บไว้ในสถานะข้อผิดพลาดระหว่างเรียกใช้ฟังก์ชันแต่ละฟังก์ชัน หากต้องการตรวจสอบข้อผิดพลาด [e: STR\_ERR\_STAT] ให้เขียนข้อความคำสั่งต่อไปนี้ คุณสามารถยืนยันข้อผิดพลาดด้วยนิพจน์ต่อไปนี้ได้

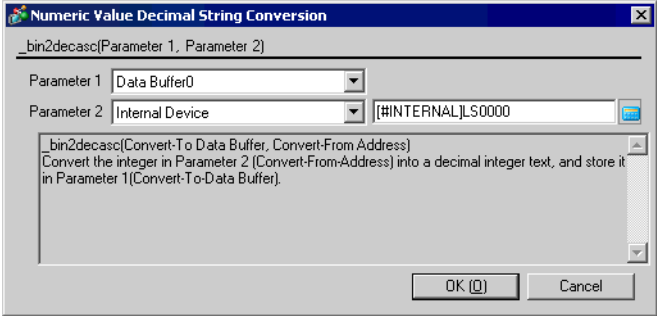
### ตัวอย่างนิพจน์

```
if ([e:STR_ERR_STAT] <> 0)           // ตรวจสอบสถานะข้อผิดพลาด
{
    set([b:[#INTERNAL]LS005000])      // กำหนดบิตของไฟสัญญาณแสดงข้อผิดพลาด
}
endif
```

#### ข้อสำคัญ

- การประมวลผลจะสิ้นสุดลงเมื่อเกิดข้อผิดพลาดและกลับไปขั้นตอนเริ่มต้นของฟังก์ชันหลัก (หากมีคำสั่งเข้ามาระหว่างฟังก์ชันกำลังทำงาน ระบบจะกลับไปบรรทัดที่เรียกฟังก์ชันนั้นขึ้นมา)

## 21.11.6 Numeric Value Decimal String Conversion

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                 |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ใช้ฟังก์ชันนี้เพื่อแปลงจำนวนเต็มให้เป็นสตริงเลขฐานสิบ โดยฟังก์ชันจะแปลงจำนวนเต็มใน Parameter 2 (ตำแหน่งที่ถูกแปลง) ให้เป็นข้อความจำนวนเต็มเลขฐานสิบ และเก็บไว้ใน Parameter 1 (บัฟเฟอร์ข้อมูลที่แปลงแล้ว)                                 |
| รูปแบบ     | <p>_bin2decasc (บัฟเฟอร์ข้อมูลที่แปลงแล้ว, [ตำแหน่งที่ถูกแปลง])</p>  <p>Parameter 1: บัฟเฟอร์ข้อมูล<br/>Parameter 2: อุปกรณ์ภายใน, ตำแหน่งชั่วคราว</p> |

### ตัวอย่างนิพจน์ 1 (เมื่อข้อมูลยาว 16 บิต)

\_bin2decasc (databuf0, [w:[#INTERNAL]LS0100])

|        |        |
|--------|--------|
|        | 16 บิต |
| LS0100 | 1234   |

ข้อมูลข้างบนจะถูกแปลงดังต่อไปนี้ โปรดสังเกตว่าระบบจะเพิ่ม “ค่าศูนย์ (0x00)” เข้าไป

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

### ตัวอย่างนิพจน์ 2 (เมื่อข้อมูลยาว 32 บิต)

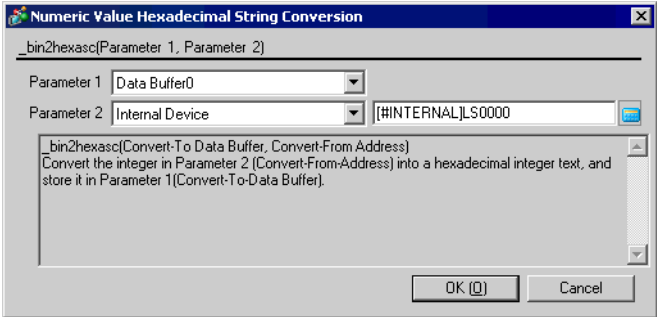
\_bin2decasc (databuf0, [w:[#INTERNAL]LS0100])

|        |          |
|--------|----------|
|        | 32 บิต   |
| LS0100 | 12345678 |
| LS0102 |          |

ข้อมูลข้างบนจะถูกแปลงดังต่อไปนี้

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 35h   | '5'      |
| databuf0[5] | 36h   | '6'      |
| databuf0[6] | 37h   | '7'      |
| databuf0[7] | 38h   | '8'      |
| databuf0[8] | 00h   | ค่าศูนย์ |

## 21.11.7 Numeric Value Decimal String Conversion

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                 |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ใช้ฟังก์ชันนี้เพื่อแปลงข้อมูลเลขฐานสองให้เป็นสตริงเลขฐานสิบหก โดยฟังก์ชันจะแปลงจำนวนเต็มใน Parameter 2 (ตำแหน่งที่ถูกแปลง) ให้เป็นข้อความจำนวนเต็มเลขฐานสิบหก และเก็บไว้ใน Parameter 1 (บัฟเฟอร์ข้อมูลที่แปลงแล้ว)                       |
| รูปแบบ     | <p>_bin2hexasc (บัฟเฟอร์ข้อมูลที่แปลงแล้ว, [ตำแหน่งที่ถูกแปลง])</p>  <p>Parameter 1: บัฟเฟอร์ข้อมูล<br/>Parameter 2: อุปกรณ์ภายใน, ตำแหน่งชั่วคราว</p> |

### ตัวอย่างนิพจน์ 1 (เมื่อข้อมูลยาว 16 บิต)

\_bin2hexasc (databuf0, [w:[#INTERNAL]LS0100])

|        |        |
|--------|--------|
|        | 16 บิต |
| LS0100 | 1234h  |

ข้อมูลข้างบนจะถูกแปลงดังต่อไปนี้ โปรตัสเกตวาระบบจะเพิ่ม “ค่าศูนย์ (0x00)” เข้าไป

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

### ตัวอย่างนิพจน์ 2 (เมื่อข้อมูลยาว 32 บิต)

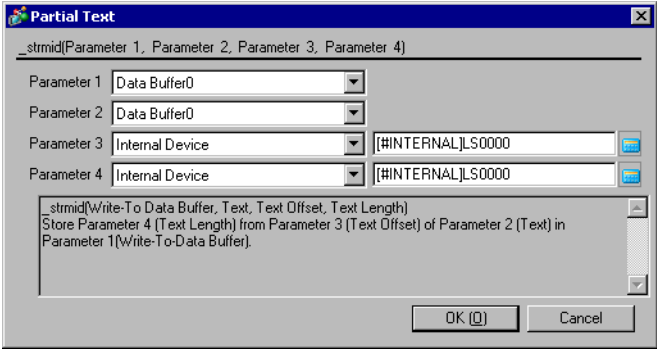
\_bin2hexasc (databuf0, [w:[#INTERNAL]LS0100])

|        |           |
|--------|-----------|
|        | 32 บิต    |
| LS0100 | 12345678h |
| LS0102 |           |

ข้อมูลข้างบนจะถูกแปลงดังต่อไปนี้

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 35h   | '5'      |
| databuf0[5] | 36h   | '6'      |
| databuf0[6] | 37h   | '7'      |
| databuf0[7] | 38h   | '8'      |
| databuf0[8] | 00h   | ค่าศูนย์ |

## 21.11.8 Partial Text

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | ค้นข้อมูลตามความยาวของสตริงจากออฟเซตที่ระบุไว้ของสตริง แล้วจัดเก็บข้อมูลไว้ในบัฟเฟอร์ข้อมูลอื่น ฟังก์ชันจะจัดเก็บ Parameter 4 (ความยาวข้อความ) จาก Parameter 3 (ออฟเซตข้อความ) ของ Parameter 2 (ข้อความ) ไว้ใน Parameter 1 (บัฟเฟอร์ข้อมูลปลายทางการเขียน)                                                                                                                                                                                                                                         |
| รูปแบบ     | <p>_strmid (บัฟเฟอร์ข้อมูลปลายทางการเขียน, ข้อความ, ออฟเซตข้อความ, ความยาวข้อความ)</p>  <p>Parameter 1: บัฟเฟอร์ข้อมูล</p> <p>Parameter 2: สตริง, บัฟเฟอร์ข้อมูล</p> <p>Parameter 3: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 3 คือตั้งแต่ 1 ถึง 1,024)</p> <p>Parameter 4: ค่าตัวเลข, อุปกรณ์ภายใน, ตำแหน่งชั่วคราว (ช่วงที่ใช้ได้สำหรับ Parameter 4 คือตั้งแต่ 0 ถึง 1,024)</p> |

### ตัวอย่างนิพจน์

\_strmid (databuf0, "12345678", 2, 4)

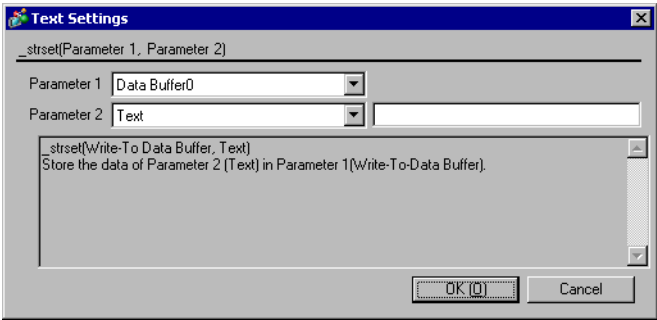
ระบบจะจัดเก็บข้อมูล 4 ไบต์ที่ค้นมาจากออฟเซต 2 ของสตริง "12345678" ไว้ใน "databuf0"

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 33h   | '3'      |
| databuf0[1] | 34h   | '4'      |
| databuf0[2] | 35h   | '5'      |
| databuf0[3] | 36h   | '6'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

#### ข้อสำคัญ

- เมื่อพยายามจะค้นสตริงที่ยาวกว่าสตริงที่ฟังก์ชัน "strmid ()" ระบุไว้ หรือเมื่อระบุค่าออฟเซตที่มากกว่าสตริงที่ระบุ ระบบจะแสดงข้อผิดพลาดหมายเลข 3 (ข้อผิดพลาดในการดึงสตริง) ของสถานะข้อผิดพลาดของสตริง [e: STR\_ERR\_STAT]
- การประมวลผลจะสิ้นสุดลงเมื่อเกิดข้อผิดพลาดและกลับไปที่ตอนเริ่มต้นของฟังก์ชันหลัก (หากมีคำสั่งเข้ามาระหว่างฟังก์ชันกำลังทำงาน ระบบจะกลับไปที่ยี่ห้อที่เรียกฟังก์ชันนั้นขึ้นมา)

## 21.11.9 Text Settings

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                      |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | จัดเก็บสตริงแบบตายตัวไว้ในบัฟเฟอร์ข้อมูล ฟังก์ชันนี้จะจัดเก็บข้อมูลของ Parameter 2 (ข้อความ) ไว้ใน Parameter 1 (บัฟเฟอร์ข้อมูลปลายทางการเขียน)                                                                                                                                                |
| รูปแบบ     | <p>_strset (บัฟเฟอร์ข้อมูลปลายทางการเขียน, ข้อความ)</p>  <p>Parameter 1: บัฟเฟอร์ข้อมูล<br/>Parameter 2: ข้อความ, ค่าตัวเลข (รหัสตัวอักษร) (ช่วงที่ใช้ได้สำหรับ Parameter 2 คือ 0 และตั้งแต่ 1 ถึง 255)</p> |

### ตัวอย่างนิพจน์

\_strset (databuf0, "ABCD")

สตริงถูกจัดเก็บไว้ในบัฟเฟอร์ข้อมูลตามที่แสดงไว้ด้านล่าง

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 41h   | 'A'      |
| databuf0[1] | 42h   | 'B'      |
| databuf0[2] | 43h   | 'C'      |
| databuf0[3] | 44h   | 'D'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

#### ข้อสำคัญ

- สามารถกำหนดสตริงได้สูงสุด 255 อักขระ หากต้องการสร้างสตริงที่ยาวกว่านี้ ให้จัดเก็บสตริงไว้ในบัฟเฟอร์อื่นและเชื่อมสตริงด้วยฟังก์ชันการเชื่อมสตริง (\_strcat)
- หากต้องการล้างบัฟเฟอร์ข้อมูล ให้สร้างสตริงเปล่า ""  
ตัวอย่าง \_strset (databuf0, "")  
\_strset (databuf0, 0)

## 21.11.10 Text Length

| รายการ     | คำอธิบาย                                                                                                                                                        |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | รับความยาวของสตริงที่จัดเก็บไว้ ฟังก์ชันนี้จะจัดเก็บความยาวของ Parameter 2 (ข้อความ) ไว้ใน Parameter 1 (ตำแหน่งที่จะเขียนความยาวข้อความ) (ไม่รวมอักขระค่าศูนย์) |
| รูปแบบ     | <p>_strlen (ตำแหน่งที่จะเขียนความยาวข้อความ, ข้อความ)</p> <p>Parameter 1: อุปกรณ์ภายใน, ตำแหน่งชั่วคราว<br/>Parameter 2: ข้อความ, บัฟเฟอร์ข้อมูล</p>            |

### ตัวอย่างนิพจน์ 1

\_strlen ([w:[#INTERNAL]LS0100], "ABCD")

เมื่อรันข้อความคำสั่งข้างบน ระบบจะเขียนความยาวของสตริงลงในพื้นที่ LS0100 ตามที่แสดงไว้ด้านล่าง

|        |   |
|--------|---|
| LS0100 | 4 |
|--------|---|

### ตัวอย่างนิพจน์ 2

\_strlen ([t:0000], databuf0)

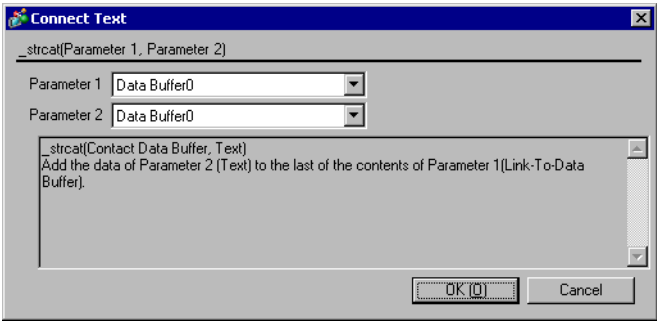
รายละเอียดของ "databuf0" มีดังนี้

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

เมื่อรันข้อความคำสั่งข้างบน ระบบจะเขียนความยาวของสตริงลงใน [t: 0000] ตามที่แสดงไว้ด้านล่าง

|       |   |
|-------|---|
| t0000 | 4 |
|-------|---|

## 21.11.11 Connect Text

| รายการ     | คำอธิบาย                                                                                                                                                                                                                                                                                                |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ข้อมูลสรุป | เชื่อมสตริงอักขระหรือรหัสอักขระด้วยบัฟเฟอร์ข้อความ ฟังก์ชันนี้จะเพิ่มข้อมูลของ Parameter 2 (ข้อความ) ไว้ที่ตอนท้ายเนื้อหาของ Parameter 1 (บัฟเฟอร์ข้อมูลการติดต่อ)                                                                                                                                      |
| รูปแบบ     | <p>_strcat (บัฟเฟอร์ข้อมูลการติดต่อ, ข้อความ)</p>  <p>Parameter 1: บัฟเฟอร์ข้อมูล<br/>Parameter 2: ข้อความ, ค่าตัวเลข (รหัสตัวอักษร), บัฟเฟอร์ข้อมูล (ช่วงที่ใช้ได้สำหรับ Parameter 2 คือ 0 และตั้งแต่ 1 ถึง 255)</p> |

### ตัวอย่างนิพจน์ 1

\_strcat (databuf0, "ABCD")

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 00h   | ค่าศูนย์ |

เมื่อเชื่อม "ABCD" เข้าด้วยกันตามที่แสดงข้างบน ผลที่ได้จะเป็นดังนี้ โปรดสังเกตว่าระบบจะเพิ่ม "ค่าศูนย์ (0x00)" เข้าไป

|             |       |          |
|-------------|-------|----------|
|             | 8 บิต |          |
| databuf0[0] | 31h   | '1'      |
| databuf0[1] | 32h   | '2'      |
| databuf0[2] | 33h   | '3'      |
| databuf0[3] | 34h   | '4'      |
| databuf0[4] | 41h   | 'A'      |
| databuf0[5] | 42h   | 'B'      |
| databuf0[6] | 43h   | 'C'      |
| databuf0[7] | 44h   | 'D'      |
| databuf0[8] | 00h   | ค่าศูนย์ |

#### ข้อควรระวัง

- สามารถกำหนดสตริงได้สูงสุด 255 อักขระ
  - หากคุณตั้งค่าสตริงเปล่า "" หรือค่าตัวเลข 0 ให้กับ Parameter 2 บัฟเฟอร์ข้อมูลของ Parameter 1 จะไม่เปลี่ยนแปลง
- ตัวอย่าง \_strcat (databuf0,"")  
\_strcat (databuf0,0)

## 21.12 ตัวอย่างการทำงาน

### 21.12.1 ตัวอย่างการทำงานของฟังก์ชันตรรกศาสตร์

#### ■ ต่อไปนี้เป็นตัวอย่างการทำงานของฟังก์ชันตรรกศาสตร์

◆  $((100 > 99) \text{ and } (200 <= 100))$

ผลลัพธ์: เปิด

◆  $((100 > 99) \text{ and } (200 <= 200))$

ผลลัพธ์: ปิด

◆  $((100 > 99) \text{ or } (200 <= 200))$

ผลลัพธ์: เปิด

◆  $((100 < 99) \text{ or } (200 <= 200))$

ผลลัพธ์: ปิด

◆  $\text{not } (100 > 99)$

ผลลัพธ์: ปิด

◆  $\text{not } (100 < 99)$

ผลลัพธ์: เปิด

◆  $[w:D200] < 10$

ผลลัพธ์: เป็นจริงหาก D200 น้อยกว่า 10

◆  $\text{not } [w:D200]$

ผลลัพธ์: เป็นจริงหาก D200 คือ 0

◆  $([w:D200] == 2) \text{ or } ([w:D200] == 5)$

ผลลัพธ์: เป็นจริงหาก D200 คือ 2 หรือ 5

◆  $([w:D200] < 5) \text{ and } ([w:D300] < 8)$

ผลลัพธ์: เป็นจริงหาก D200 น้อยกว่า 5 และ D300 น้อยกว่า 8

◆  $[w:D200] < 10$

ผลลัพธ์: เป็นจริงหาก D200 น้อยกว่า 10

◆  $\text{not } [w:D200]$

ผลลัพธ์: เป็นจริงหาก D200 คือ 0

◆  $([w:D200] == 2) \text{ or } ([w:D200] == 5)$

ผลลัพธ์: เป็นจริงหาก D200 คือ 2 หรือ 5

◆  $([w:D200] < 5) \text{ and } ([w:D300] < 8)$

ผลลัพธ์: เป็นจริงหาก D200 น้อยกว่า 5 และ D300 น้อยกว่า 8

## 21.12.2 ตัวอย่างการทำงานของบิต

## ■ ต่อไปนี้เป็นตัวอย่างการทำงานของบิต

## ◆ [ w:D200 ] &lt;&lt; 4

ผลลัพธ์: ข้อมูลใน D200 จะถูกเลื่อนไปทางซ้าย 4 บิต

## ◆ [ w:D200 ] &gt;&gt; 4

ผลลัพธ์: ข้อมูลใน D200 จะถูกเลื่อนไปทางขวา 4 บิต

## ◆ จัดเก็บ 12(0000Ch) ไว้ใน D301 โดยใช้รูปแบบ BIN

[ w:D200 ] = [ w:D300 ] >> [ w:D301 ]

ผลลัพธ์: ข้อมูลใน D300 ถูกเลื่อนไปทางขวา 12 บิตและกำหนดให้ตำแหน่ง D200

## ◆ [ w:D200 ] &lt;&lt; 4

ผลลัพธ์: ข้อมูลใน D200 จะถูกเลื่อนไปทางซ้าย 4 บิต

## ◆ [ w:D200 ] &gt;&gt; 4

ผลลัพธ์: ข้อมูลใน D200 จะถูกเลื่อนไปทางขวา 4 บิต

## ◆ จัดเก็บ 12(0000Ch) ไว้ใน D310 โดยใช้รูปแบบ BIN

[ w:D200 ] = [ w:D300 ] >> [ w:D310 ]

ผลลัพธ์: ข้อมูลใน D300 ถูกเลื่อนไปทางขวา 12 บิตและกำหนดให้ตำแหน่ง D200

## ◆ Bit AND

0 & 0 ผลลัพธ์: 0

0 & 1 ผลลัพธ์: 0

1 & 1 ผลลัพธ์: 1

0x1234 & 0xF0F0 ผลลัพธ์: 0x1030

## ◆ Bit OR

0 | 0 ผลลัพธ์: 0

0 | 1 ผลลัพธ์: 1

1 | 1 ผลลัพธ์: 1

0x1234 | 0x9999 ผลลัพธ์: 0x9BBF

## ◆ Bit XOR

0 ^ 0 ผลลัพธ์: 0

0 ^ 1 ผลลัพธ์: 1

1 ^ 1 ผลลัพธ์: 0

## ◆ Bit 1's complement (NOT) (เมื่อรูปแบบข้อมูลคือ Bin16 +)

NOT 0 ผลลัพธ์: 0xFFFF

NOT 1 ผลลัพธ์: 0xFFFE

## 21.12.3 ตัวอย่างการใช้แบรนช์ตามเงื่อนไข

## ■ ความคุมแบรนช์โดยใช้นิพจน์ “if-endif” และ “if-else-endif”

## ◆ if - endif

if (เงื่อนไข)

{ กระบวนการ 1 }

endif

หากเงื่อนไขเป็นจริง จะดำเนินการกระบวนการ 1 หากเป็นเท็จ ระบบจะไม่สนใจกระบวนการ 1

ตัวอย่าง

if ( [ w:D200 ] &lt; 5 )

{

[ w:D100 ] = 1

}

endif

หากข้อมูลใน D200 น้อยกว่า 5 จะกำหนดค่า 1 ในตำแหน่ง D100

## ◆ if - else - endif

if (เงื่อนไข)

{ กระบวนการ 1 }

else

{ กระบวนการ 2 }

endif

หากเงื่อนไขเป็นจริง จะดำเนินการกระบวนการ 1 หากเป็นเท็จ จะดำเนินการกระบวนการ 2

ตัวอย่าง

if ( [ w:D200 ] &lt; 5 )

{

[ w:D100 ] = 1

}

else

{

[ w:D100 ] = 0

}

endif

หากข้อมูลใน D200 น้อยกว่า 5 จะแทนตำแหน่ง D100 ด้วยค่า 1 หากมากกว่า 5 จะแทนตำแหน่ง D100 ด้วยค่า 0

## 21.12.4 ตัวอย่างการใช้ตำแหน่งออฟเซต

## ■ ตัวอย่างการคำนวณพิเศษโดยใช้การระบุออฟเซต [w:D00100]#[t:0000] จะเป็นดังนี้

- ◆ เมื่อการตั้งค่าสคริปต์มีความยาว 16 บิตและไม่มีเครื่องหมาย โดยมี [t:0000] = 65526 และตำแหน่งที่ระบุคือ [w:D00090]

$$100 + 65526 = 64(\text{Hex}) + \text{FFF6}(\text{Hex}) = \text{1005A}(\text{Hex}) \rightarrow 005A(\text{Hex}) = 90$$

↑  
บิตที่ใช้ได้คือ 16 บิตล่าง

- ◆ เมื่อการตั้งค่าสคริปต์มีความยาว 16 บิตและมีเครื่องหมาย โดยมี [t:0000] = -10 และตำแหน่งที่ระบุคือ [w:D00090]

$$100 + (-10) = 64(\text{Hex}) + \text{FFF6}(\text{Hex}) = \text{1005A}(\text{Hex}) \rightarrow 005A(\text{Hex}) = 90$$

↑  
บิตที่ใช้ได้คือ 16 บิตล่าง

- ◆ เมื่อการตั้งค่าสคริปต์มีความยาว 32 บิตและไม่มีเครื่องหมาย โดยมี [t:0000] = 4294901840 และตำแหน่งที่ระบุคือ [w:D00180]

$$100 + 4294901840 = 64(\text{Hex}) + \text{FFFF0050}(\text{Hex}) = \text{FFFF00B4}(\text{Hex}) \rightarrow 00B4(\text{Hex}) = 180$$

↑  
บิตที่ใช้ได้คือ 16 บิตล่าง

- ◆ เมื่อการตั้งค่าสคริปต์มีความยาว 32 บิตและมีเครื่องหมาย โดยมี [t:0000] = -65456 และตำแหน่งที่ระบุคือ [w:D00180]

$$100 + (-65456) = 64(\text{Hex}) + \text{FFFF0050}(\text{Hex}) = \text{FFFF00B4}(\text{Hex}) \rightarrow 00B4(\text{Hex}) = 180$$

↑  
บิตที่ใช้ได้คือ 16 บิตล่าง

**ข้อสำคัญ**

- ระบบจะถือว่าตำแหน่งออฟเซตเป็นข้อมูลแบบ Bin 16 บิตเสมอ ไม่ว่าจะตั้งค่าความยาวบิต และชนิดข้อมูลของสคริปต์ไว้เช่นไรก็ตาม หากผลการดำเนินการเกินกว่า 16 บิต (ค่าสูงสุด: 65535) ระบบจะถือว่าบิต 1 ถึงบิต 15 เป็นบิตที่ถูกตัด โดยไม่สนใจบิต 16 และบิตอื่นๆ

## 21.13 รายการคำสั่ง

| รายการ         | คำสั่ง/ฟังก์ชัน                | D-Script/<br>Global D-Script | Extended Script    |
|----------------|--------------------------------|------------------------------|--------------------|
| ชนิดข้อมูล     | Bin, BCD                       | รองรับ                       | เฉพาะ Bin เท่านั้น |
| ความยาวบิต     | 16 บิต, 32 บิต                 | รองรับ                       | รองรับ             |
| เครื่องหมาย    | มีเครื่องหมาย/ไม่มีเครื่องหมาย | รองรับ                       | รองรับ             |
| ทริกเกอร์      | การตั้งค่าตัวตั้งเวลา          | รองรับ                       | ไม่รองรับ          |
|                | ขอบขาขึ้นของบิต                | รองรับ                       | ไม่รองรับ          |
|                | ขอบขาลงของบิต                  | รองรับ                       | ไม่รองรับ          |
|                | การทำงานเป็นคู่ของบิต          | รองรับ                       | ไม่รองรับ          |
|                | เงื่อนไขของบิตที่เป็นจริง      | รองรับ                       | ไม่รองรับ          |
|                | เงื่อนไขของบิตที่เป็นเท็จ      | รองรับ                       | ไม่รองรับ          |
| วาด            | Call Screen                    | รองรับ                       | ไม่รองรับ          |
|                | Dot                            | รองรับ                       | รองรับ             |
|                | Line                           | รองรับ                       | รองรับ             |
|                | Circle                         | รองรับ                       | รองรับ             |
|                | Rectangle                      | รองรับ                       | รองรับ             |
| ตัวดำเนินการ   | Addition (+)                   | รองรับ                       | รองรับ             |
|                | Subtraction (—)                | รองรับ                       | รองรับ             |
|                | Margin (%)                     | รองรับ                       | รองรับ             |
|                | Multiplication (*)             | รองรับ                       | รองรับ             |
|                | Division (/)                   | รองรับ                       | รองรับ             |
|                | Assignment (=)                 | รองรับ                       | รองรับ             |
| การเปรียบเทียบ | Logical AND                    | รองรับ                       | รองรับ             |
|                | Logical OR                     | รองรับ                       | รองรับ             |
|                | Negation (NOT)                 | รองรับ                       | รองรับ             |
|                | less than (<)                  | รองรับ                       | รองรับ             |
|                | less than or equal to (<=)     | รองรับ                       | รองรับ             |
|                | not equal to (<>)              | รองรับ                       | รองรับ             |
|                | more than (>)                  | รองรับ                       | รองรับ             |
|                | more than or equal to (>=)     | รองรับ                       | รองรับ             |
|                | Equivalent (=)                 | รองรับ                       | รองรับ             |

ต่อ

| รายการ                 | คำสั่ง/ฟังก์ชัน                                             | D-Script/<br>Global D-Script | Extended Script      |
|------------------------|-------------------------------------------------------------|------------------------------|----------------------|
| การทำงานของหน่วยความจำ | Copy Memory: memcpy ( )                                     | รองรับ                       | รองรับ               |
|                        | Initialize Memory: memset ( )                               | รองรับ                       | รองรับ               |
|                        | Copy Memory(Variable Specification)<br>_memcpy_EX ( )       | รองรับ                       | รองรับ               |
|                        | Initialize Memory(Variable Specification)<br>_memset_EX ( ) | รองรับ                       | รองรับ               |
|                        | Offset Address                                              | รองรับ                       | รองรับ               |
|                        | Shift Memory                                                | รองรับ                       | รองรับ               |
|                        | Ring Shift Memory                                           | รองรับ                       | รองรับ               |
|                        | Search Memory                                               | รองรับ                       | รองรับ               |
|                        | Compare Memory                                              | รองรับ                       | รองรับ               |
| การทำงานของบิต         | Left Shift (<<)                                             | รองรับ                       | รองรับ               |
|                        | Right Shift (>>)                                            | รองรับ                       | รองรับ               |
|                        | Bit-wise AND (&)                                            | รองรับ                       | รองรับ               |
|                        | Bit-wise OR (!)                                             | รองรับ                       | รองรับ               |
|                        | Bit-wise XOR (^)                                            | รองรับ                       | รองรับ               |
|                        | 1's Complement                                              | รองรับ                       | รองรับ               |
|                        | Set Bit: set ( )                                            | รองรับ                       | รองรับ               |
|                        | Clear Bit: clear ( )                                        | รองรับ                       | รองรับ               |
|                        | Toggle Bit: toggle ( )                                      | รองรับ                       | รองรับ               |
| นิพจน์คำอธิบาย         | if ( )                                                      | รองรับ                       | รองรับ               |
|                        | if ( ) else                                                 | รองรับ                       | รองรับ               |
|                        | loop ( ), break                                             | รองรับ                       | รองรับ               |
|                        | loop ( ) infinite loop                                      | ไม่รองรับ                    | รองรับ               |
| ตำแหน่ง                | ตำแหน่งบิต                                                  | รองรับ                       | อุปกรณ์ภายใน         |
|                        | ตำแหน่งเวิร์ด                                               | รองรับ                       | อุปกรณ์ภายใน         |
|                        | ตำแหน่งเวิร์ดชั่วคราว                                       | รองรับรองรับ                 | รองรับ <sup>*1</sup> |
| ค่าคงที่               | Dec, Hex, Oct                                               | รองรับ                       | รองรับ               |

ต่อ

| รายการ                    | คำสั่ง/ฟังก์ชัน                                                   | D-Script/<br>Global D-Script | Extended Script |
|---------------------------|-------------------------------------------------------------------|------------------------------|-----------------|
| ฟังก์ชัน<br>SIO           | Receive: IO_READ ([p:SIO])                                        | รองรับ                       | รองรับ          |
|                           | Send: IO_WRITE ([p:SIO])                                          | รองรับ                       | รองรับ          |
|                           | Extended Receive<br>_IO_READ_EX ( )                               | ไม่รองรับ                    | รองรับ          |
|                           | Extended Send<br>_IO_WRITE_EX ( )                                 | ไม่รองรับ                    | รองรับ          |
|                           | ฟังก์ชัน<br>Standby Reception<br>_IO_READ_WAIT ( )                | ไม่รองรับ                    | รองรับ          |
|                           | ตัวแปรควบคุม<br>[c:EXT_SIO_CTRL]                                  | รองรับ                       | รองรับ          |
|                           | สถานะ<br>[s:EXT_SIO_STAT]                                         | รองรับ                       | รองรับ          |
|                           | จำนวน ข้อมูลที่ได้รับ<br>[r:EXT_SIO_RCV]                          | รองรับ                       | รองรับ          |
|                           | ฟังก์ชัน Standby: _wait ( )                                       | ไม่รองรับ                    | รองรับ          |
| การทำงานของ<br>ของข้อความ | ข้อความ                                                           | ไม่รองรับ                    | รองรับ          |
|                           | บัฟเฟอร์ข้อมูล:<br>databuf0, databuf1,<br>databuf2, databuf3      | ไม่รองรับ                    | รองรับ          |
|                           | Text Settings<br>_strset ( )                                      | ไม่รองรับ                    | รองรับ          |
|                           | From Data Buffer to Internal<br>Device<br>_dlcopy ( )             | ไม่รองรับ                    | รองรับ          |
|                           | From Internal Device To Data<br>Buffer ldcopy ( )                 | ไม่รองรับ                    | รองรับ          |
|                           | Hexadecimal Text-To-Integer<br>Conversion<br>_hexasc2bin ( )      | ไม่รองรับ                    | รองรับ          |
|                           | Decimal Text-To-Integer<br>Conversion<br>_decasc2bin ( )          | ไม่รองรับ                    | รองรับ          |
|                           | Numeric Value Hexadecimal<br>String Conversion<br>_bin2hexasc ( ) | ไม่รองรับ                    | รองรับ          |

ต่อ

| รายการ                         | คำสั่ง/ฟังก์ชัน                                               | D-Script/<br>Global D-Script | Extended Script |
|--------------------------------|---------------------------------------------------------------|------------------------------|-----------------|
| การทำงานของ<br>ของข้อความ      | Numeric Value Decimal<br>String Conversion<br>_bin2decasc ( ) | ไม่รองรับ                    | รองรับ          |
|                                | Text Length<br>_strlen ( )                                    | ไม่รองรับ                    | รองรับ          |
|                                | Connect Text<br>_strcat ( )                                   | ไม่รองรับ                    | รองรับ          |
|                                | Partial Text<br>_strmid ( )                                   | ไม่รองรับ                    | รองรับ          |
|                                | Status<br>[e: STR_ERR_STAT]                                   | ไม่รองรับ                    | รองรับ          |
| ฟังก์ชัน                       | Call                                                          | รองรับ                       | รองรับ          |
|                                | return                                                        | ไม่รองรับ                    | รองรับ          |
| ใช้งานไฟล์<br>ในการ์ด CF       | Read CSV File                                                 | รองรับ                       | รองรับ          |
|                                | Output File List<br>_CF_dir ( )                               | รองรับ                       | รองรับ          |
|                                | Read File<br>_CF_read ( )                                     | รองรับ                       | รองรับ          |
|                                | Write File<br>_CF_write ( )                                   | รองรับ                       | รองรับ          |
|                                | Delete File<br>_CF_delete ( )                                 | รองรับ                       | รองรับ          |
|                                | Change File Name<br>_CF_rename ( )                            | รองรับ                       | รองรับ          |
| การทำงานของ<br>ของเครื่องพิมพ์ | ส่งข้อมูลจากพอร์ต COM<br>IO_WRITE ([p:PRN])                   | รองรับ                       | รองรับ          |
| ดีบั๊ก                         | _debug ( )                                                    | รองรับ                       | รองรับ          |

\*1 ตำแหน่งชั่วคราวมีอยู่โดยไม่ขึ้นกับ D-Scripts และ Global D-Scripts ทั้งหมด

---

# บันทึก