

# 21

# Usare gli script

## (Programmazione senza parti)

Questo capitolo illustra come utilizzare GP-Pro EX per "Programmare senza parti" e come creare script.

Iniziare leggendo "21.1 Menu Impostazioni" (pagina 21-2), quindi andare alla pagina corrispondente.

21.1	Menu Impostazioni .....	21-2
21.2	Operazioni condizionali .....	21-6
21.3	Copiare dati in blocchi .....	21-12
21.4	Visualizzare un allarme quando si verifica un errore .....	21-16
21.5	Comunicazione con periferiche non supportate .....	21-20
21.6	Riferimento ad altri script .....	21-37
21.7	Creare script .....	21-41
21.8	Configurazione condizione trigger .....	21-46
21.9	Guida alle impostazioni .....	21-53
21.10	Limitazioni .....	21-59
21.11	Comandi programma/Espressioni condizionali .....	21-69

## 21.1 Menu Impostazioni

Gli utenti possono utilizzare i D-Script per creare programmi semplici. Usando questa funzione si possono eseguire operazioni nell'unità GP o comunicare tra l'unità GP e le periferiche non supportate.

---

### **AVVISO**

---

Assicurarsi di non utilizzare D-Script/D-Script globali per controllare sistemi che possono causare azioni pericolose per la vita oppure in grado di creare gravi lesioni.

---

**NOTA**

- I D-Script sono configurati su uno schermo di base. Lo schermo di base osserva le condizioni mentre viene visualizzato ed esegue lo script.
- Quando l'unità GP è in esecuzione, sarà eseguito un D-script globale sulla base del trigger, indipendentemente dallo schermo visualizzato.
- Gli script estesi dovranno essere utilizzati per programmi di comunicazione ad alto livello.
- Oltre agli script, si possono utilizzare programmi logici per le applicazioni di controllo.

 "28.1 Menu Impostazioni" (pagina 28-2)

---

### Operazioni condizionali

Creare uno script che cambia automaticamente gli schermi nello schermo numero 7 dopo 3 secondi.

D100=1

D100 non è 3, pertanto la parte dopo "if" non viene eseguita.

D100=2

D100 non è 3, pertanto la parte dopo "if" non viene eseguita.

D100=3

D100 = 3 la condizione è vera e [w:LS0008]=7 esegue.

➡ Procedura di configurazione (pagina 21-7)

➡ Introduzione (pagina 21-6)

### Copiare dati in blocchi

Creare uno script che rileva l'ascesa (da 0 a 1) degli indirizzi bit M0100 e copia i dati memorizzati nel dispositivo collegato in un altri indirizzo.

D0099

C
·
·
·
B
A

D0000

D0200

C
·
·
·
B
A

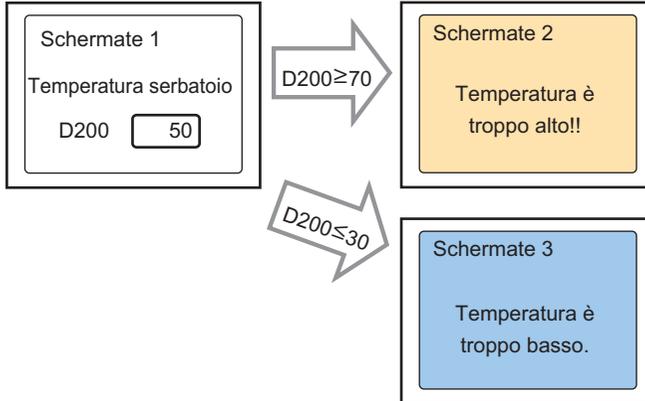
D0101

➡ Procedura di configurazione (pagina 21-13)

➡ Introduzione (pagina 21-12)

**Visualizzare un allarme quando si verifica un errore**

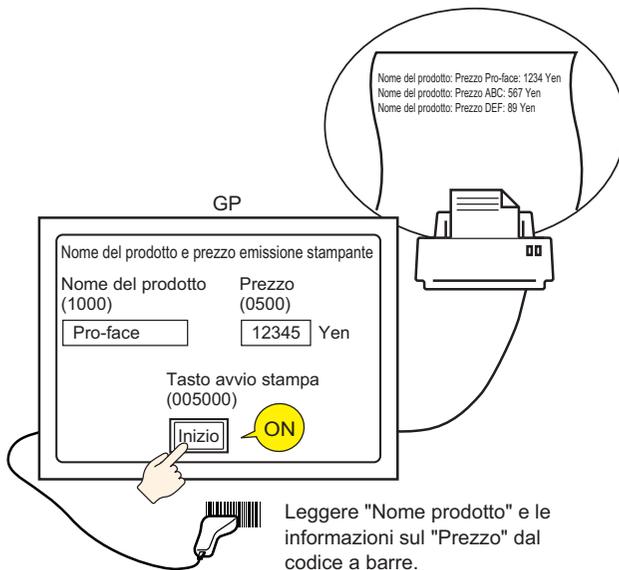
Il sistema di gestione temperatura rileva un bit errore dal dispositivo collegato e visualizza messaggi di allarme quando l'indirizzo di memorizzazione delle informazioni sulla temperatura (D200) arriva a 70°C oppure oltre, o scende al di sotto dei 30°C o a una temperatura inferiore. Inoltre, lo script conterà il numero degli errori rilevati.



- ☞ Procedura di configurazione (pagina 21-17)
- ☞ Introduzione (pagina 21-16)

**Comunicazione con periferiche non supportate**

Creare uno script esteso in grado di emettere la lettura dei dati da un codice a barre collegato all'USB a una stampante seriale collegata a COM 1.



- ☞ Procedura di configurazione (pagina 21-33)
- ☞ Introduzione (pagina 21-20)

### Riferimento ad altri script

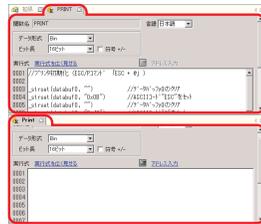
Nell finestra di dialogo [D-Script], dividere lo schermo in due schermi in senso orizzontale o verticale.

Fare clic sulle schede si passa da uno schermo all'altro.

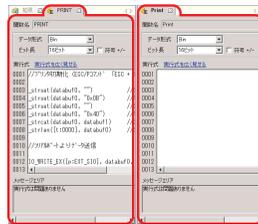


Visualizzare/modificare i due schermi affiancati

Affiancare in orizzontale



Affiancare in verticale



- ➡ Procedura di funzionamento (pagina 21-38)
- ➡ Introduzione (pagina 21-37)

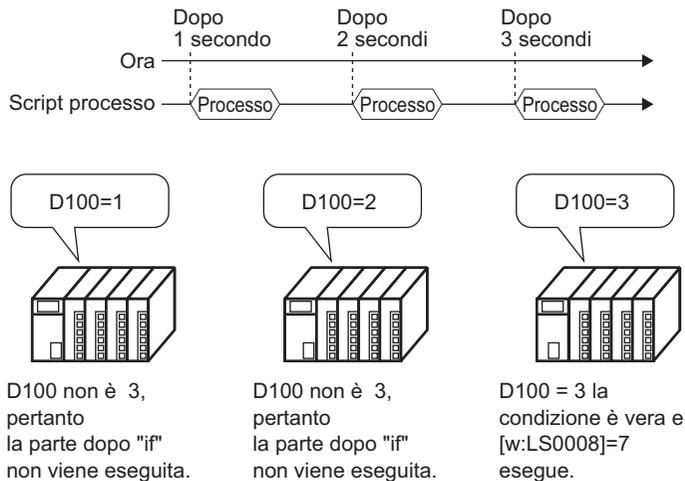
## 21.2 Operazioni condizionali

**NOTA**

- Fare riferimento alla guida per le impostazioni per altri dettagli.
  - ☞ "21.9.1 Guida alle impostazioni D-Script/[D-Script globale] comune" (pagina 21-53)
- Vedere quanto segue per altre informazioni sui comandi disponibili per gli script.
  - ☞ "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)

### Azione

Creare uno script che cambia automaticamente gli schermi nello schermo numero 7 dopo 3 secondi.



### Comandi utilizzati

Comando	Riepilogo funzioni
Assegnazione (=)	Assegnare il valore sulla parte destra alla parte sinistra. ☞ "21.11.10 Operatore" (pagina 21-154)
Aggiunta (+)	Aggiunge una costante ai dati di un dispositivo word. ☞ "21.11.10 Operatore" (pagina 21-154)
if ( )	Quando una condizione diventa vera, il processo che segue l'istruzione "if ( )" verrà eseguito. ☞ "21.11.8 Espressioni condizionali" (pagina 21-147)
Equivalente (==)	Confronta i valori sui lati destro e sinistro. Diventa "vero" se il lato sinistro corrisponde al lato destro. ☞ "21.11.9 Confronto" (pagina 21-152)
LS0008	Modifica il numero di schermi memorizzato in questo valore. ☞ "A.1.4.2 Area dati di sistema" (pagina A-11)

## Trigger

Selezionare il timer e configurare le [Impostazioni timer] su 1 secondo.



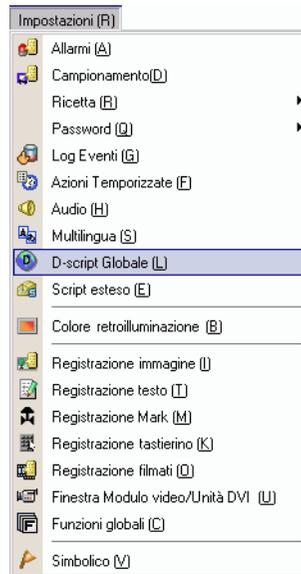
## Script completato

```

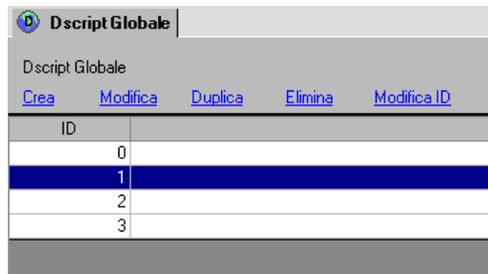
Espressione      Ingrandisci espressione  Inserisci Indirizzo
0001 [w: [PLC1]D00100]=[w: [PLC1]D00100]+1
0002 if ([w: [PLC1]D00100]==3)
0003 {
0004     [w: [#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
0008
0009
    
```

## Procedura di creazione

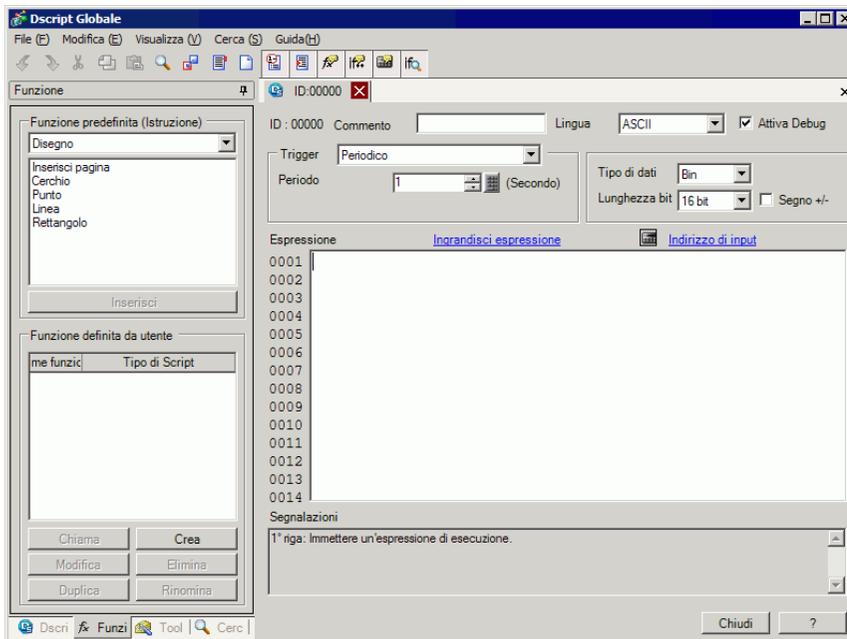
1 Nel menu [Impostazioni comuni (R)], selezionare [Impostazioni D-Script Globale (L)].



2 Fare clic su [Creare]. Per vedere uno script esistente, selezionare il numero di ID e fare clic su [Modifica], oppure fare doppio clic sulla riga del numero dell'ID.

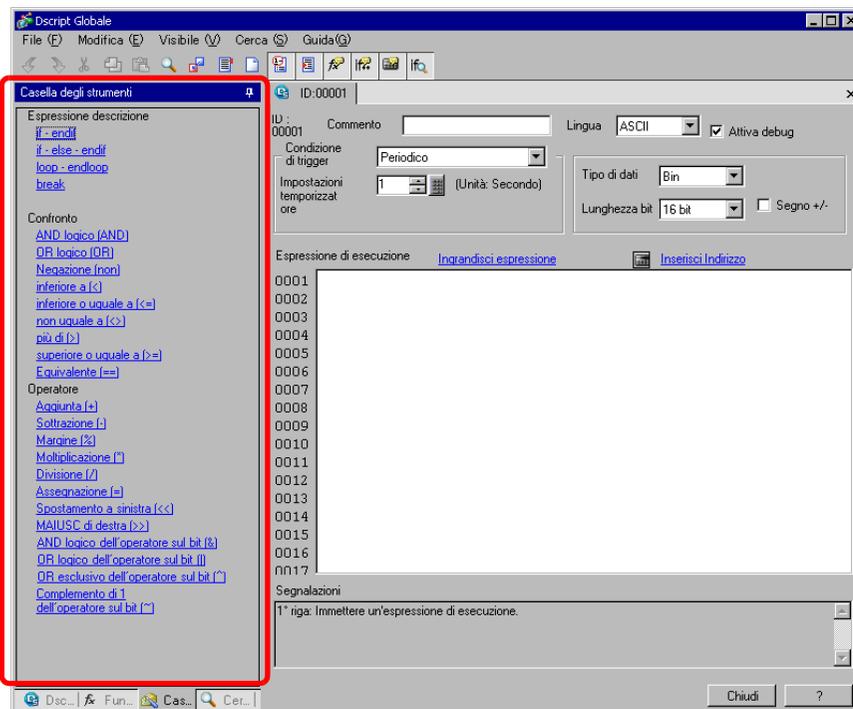


3 Viene visualizzata la finestra di dialogo [D-Script].



4 In [Trigger], selezionare [Timer] e specificare le [Impostazioni timer ] su 1 secondo.

5 Fare clic sulla scheda [Toolbox]. Il Toolbox consente di collocare facilmente un comando da utilizzare nello script.



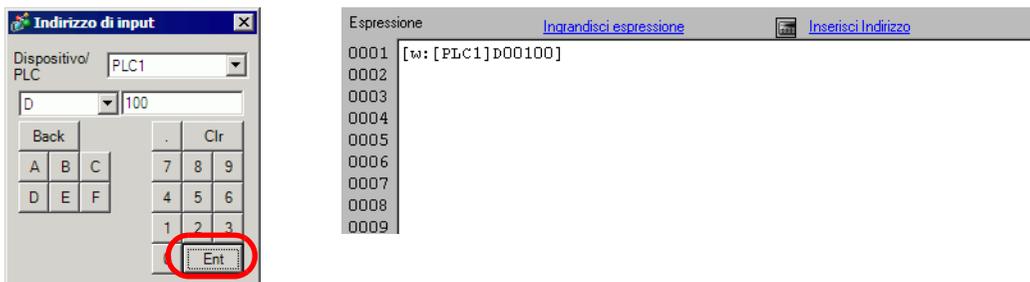
6 Creare la prima linea dello script. Se si specifica che il valore predefinito D00100 è 0, l'operazione con la prima linea sarà un'operazione di conteggio che aumenta e memorizza il

conteggio ogni volta che un processo viene completato.

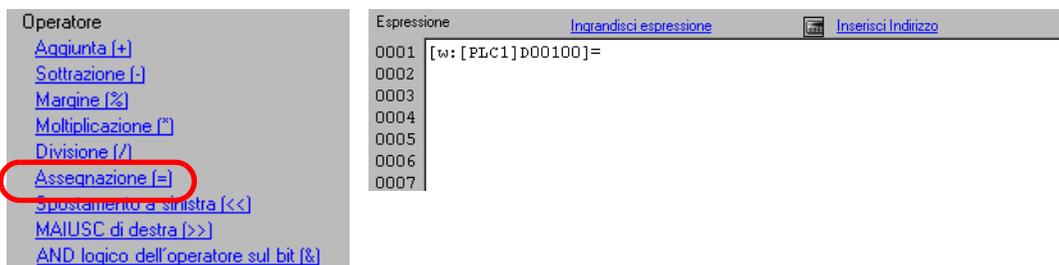
Fare clic su  e selezionare [Indirizzo word], quindi fare clic su .



7 Inserire D00100, e fare clic su [ENT]. Fare clic su [OK] nella finestra di dialogo [Indirizzo di input].



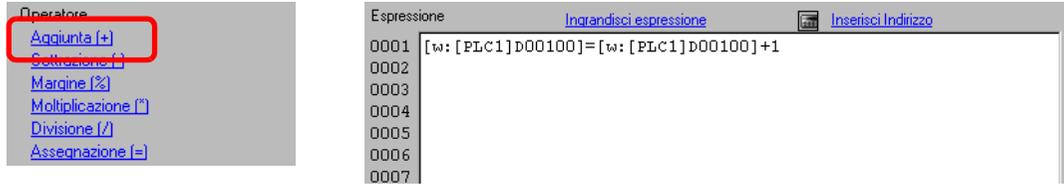
8 Fare clic su [Assegnazione (=)] nel Toolbox.



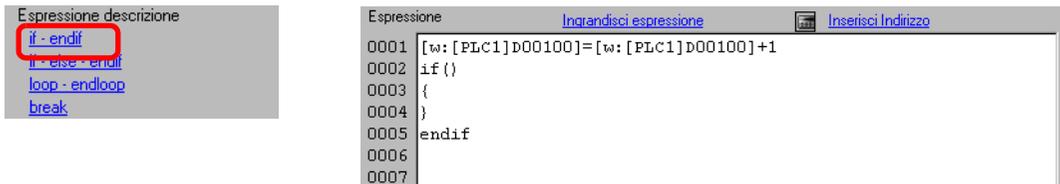
9 Collocare D00100 nello stesso modo illustrato nei passaggi 6 e 7.



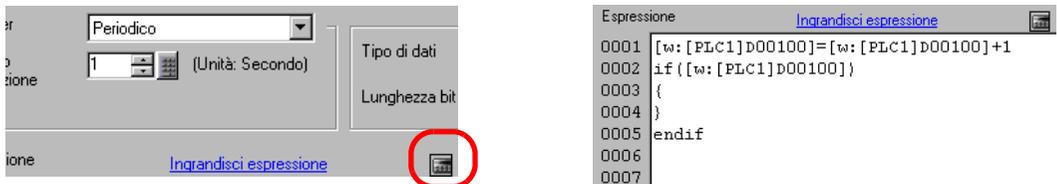
10 Fare clic su [Somma (+)] e digitare "1". La prima fila è adesso completa.



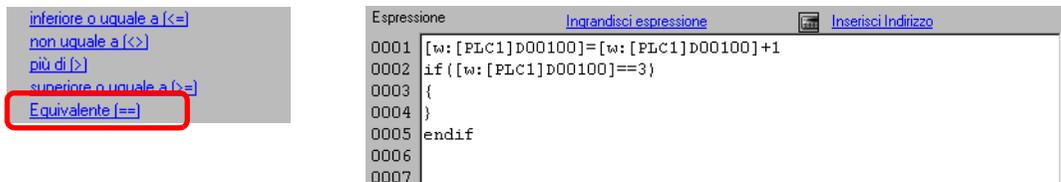
11 Creare la seconda riga dello script. Nella seconda riga, quando una condizione diventa autentica, verrà eseguito il processo che segue l'istruzione "if ( )". Fare clic su [if - endif].



12 Dopo "if", creare l'espressione condizionale tra parentesi "( )". L'espressione condizionale confronta il valore memorizzato in D00100 con "3", e diventa vera se sono uguali. Collocare il cursore tra le parentesi "( )" e ripetere i passaggi 6 e 7 per collocare un altro D00100.



13 Fare clic su [Equivalente (==)] e inserire "3". Ora la seconda fila è completa.

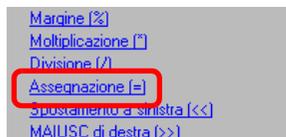


14 Collocare il cursore nelle parentesi graffe "{ }" e premere Invio (Enter). Ripetere i passaggi da 6 a 7 per collocare un altro LS0008.

```

Espressione
Ingrandisci espressione
0001 [w:[PLC1]D00100]=[w:[PLC1]D00100]+1
0002 if {[w:[PLC1]D00100]==3}
0003 {
0004     [w:[#INTERNAL]LS0008]
0005 }
0006 endif
0007
    
```

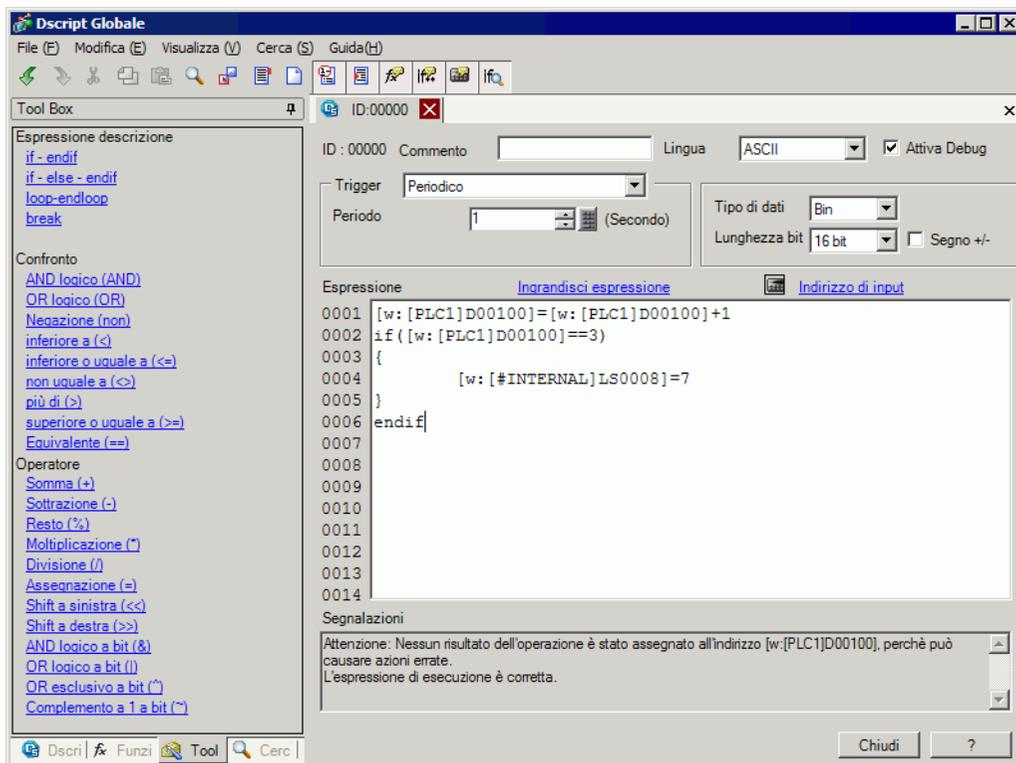
15 Fare clic su [Assegnazione (=)] ed inserire "7".



```

Espressione
Ingrandisci espressione
Inserisci indirizzo
0001 [w:[PLC1]D00100]=[w:[PLC1]D00100]+1
0002 if {[w:[PLC1]D00100]==3}
0003 {
0004     [w:[#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
    
```

16 Lo script è completo.



**NOTA**

- Quando si seleziona testo, premere i tasti [CTRL] + [MAIUSC] + [Freccia a destra]/[Freccia a sinistra] per selezionare un intero blocco di testo.
- Premere il Tasto [CTRL] + [F4] per chiudere il visualizzatore attualmente selezionato.
- Premere il tasto [Esc] per sovrascrivere lo script o per eliminarlo e uscire.

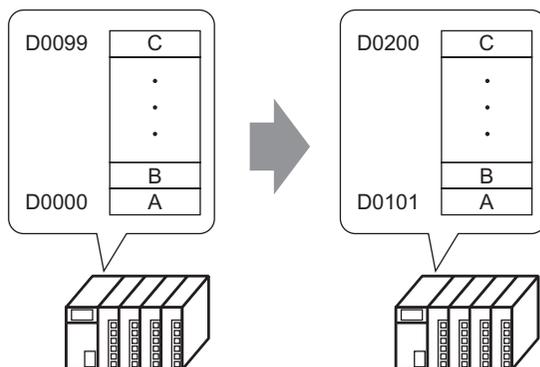
## 21.3 Copiare dati in blocchi

**NOTA**

- Fare riferimento alla guida per le impostazioni per altri dettagli.  
 ➔ "21.9.1 Guida alle impostazioni D-Script/[D-Script globale] comune" (pagina 21-53)
- Vedere quanto segue per altre informazioni sui comandi disponibili per gli script.  
 ➔ "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)

### Azione

Creare uno script che rileva l'ascensione (da 0 a 1) degli indirizzi bit M0100 e copia i dati memorizzati nel dispositivo collegato in un altri indirizzo.

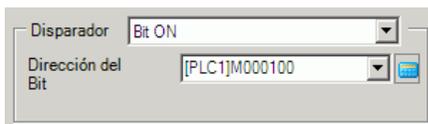


### Comandi utilizzati

Comando	Riepilogo funzioni
<b>Copia memoria memcpy ( )</b>	<p>Copia un valore memorizzato in un dispositivo in una sola operazione. I dati per il numero di indirizzi saranno copiati agli indirizzi word di destinazione copia, iniziando dal primo indirizzo word dei dati di origine.</p> <p>[Formatta]                      memcpy ([Copia nell'indirizzo], [Copia dall'indirizzo], Word)</p> <p>➔ "21.11.3 Operazione memoria" (pagina 21-79)</p>

### Trigger

In [Trigger], selezionare [Bit su ON], e impostare I[Indirizzo bit] su M000100.



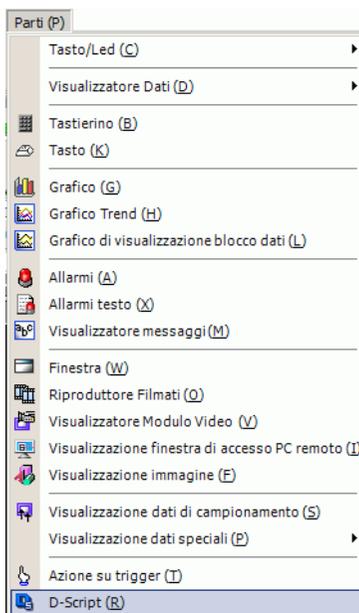
### Script completato

```

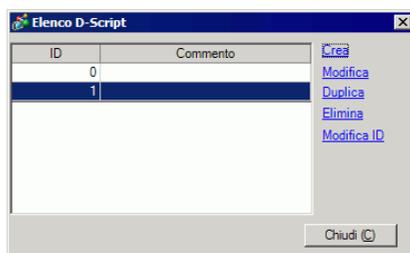
Espressione      Ingrandisci espressione      Inserisci Indirizzo
0001 memcpy([w:[PLC1]D00101], [w:[PLC1]D00000], 100)
0002
0003
0004
0005
0006
0007
0008
    
```

## Procedura di creazione

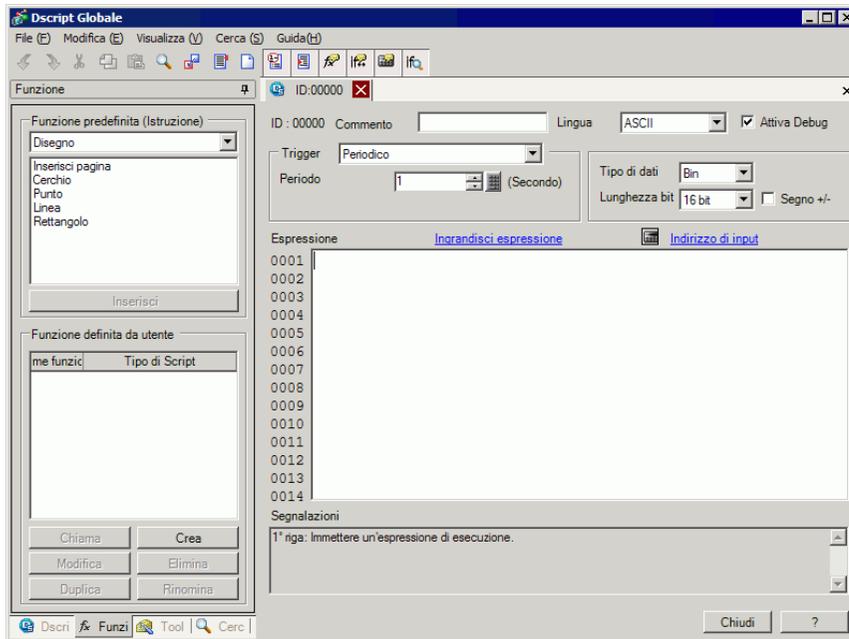
1 Nel menu [Parti (P)], selezionare [D-Script (R)] o fare clic su  dalla barra strumenti.



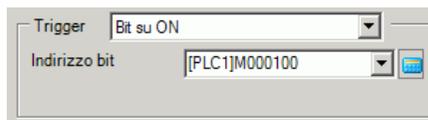
2 Fare clic su [Creare]. Gli ID per script esistenti sono visualizzati nell'[Elenco D-Script].



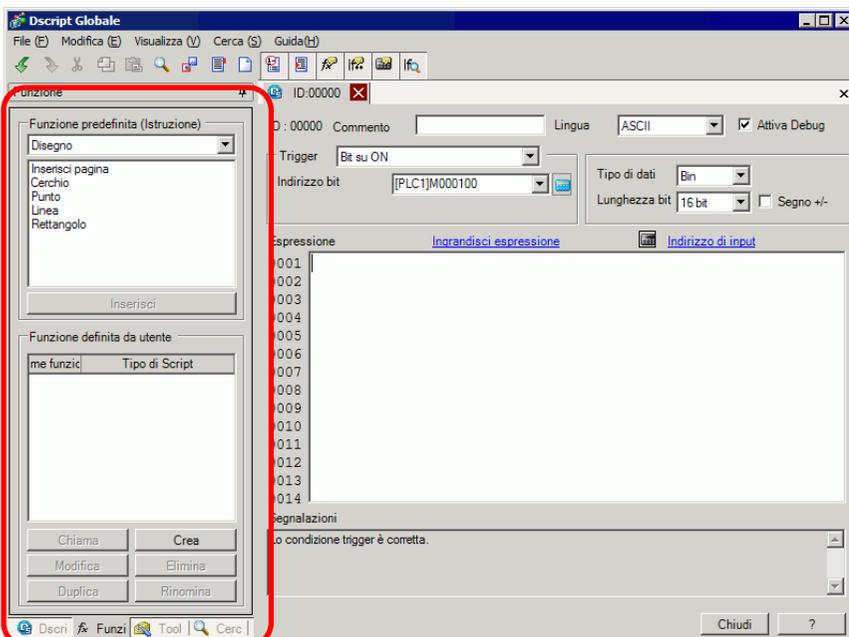
### 3 Viene visualizzata la finestra di dialogo [D-Script].



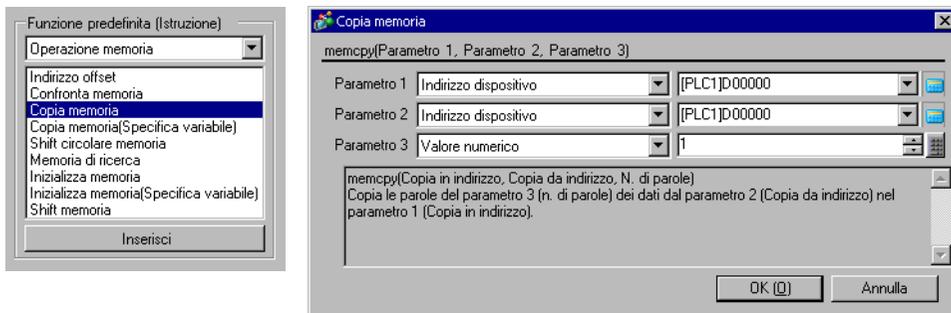
### 4 Selezionare [Bit ON] in [Trigger] e specificare M000100 come [Indirizzo bit].



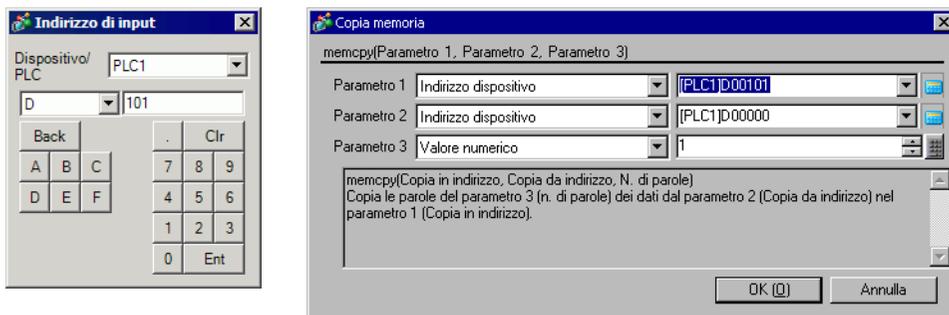
### 5 Fare clic sulla scheda [Funzione]. Le funzioni incorporate consentono di collocare facilmente un comando da usare nello script.



- 6 Selezionare [Funzione incorporata (Istruzione)] dal menu a discesa [Funzionamento memoria]).
- 7 Fare doppio clic su [Copia memoria], quindi nella finestra di dialogo successiva definire i parametri per l'indirizzo di destinazione, l'indirizzo di origine e il numero di word. Fare clic su 

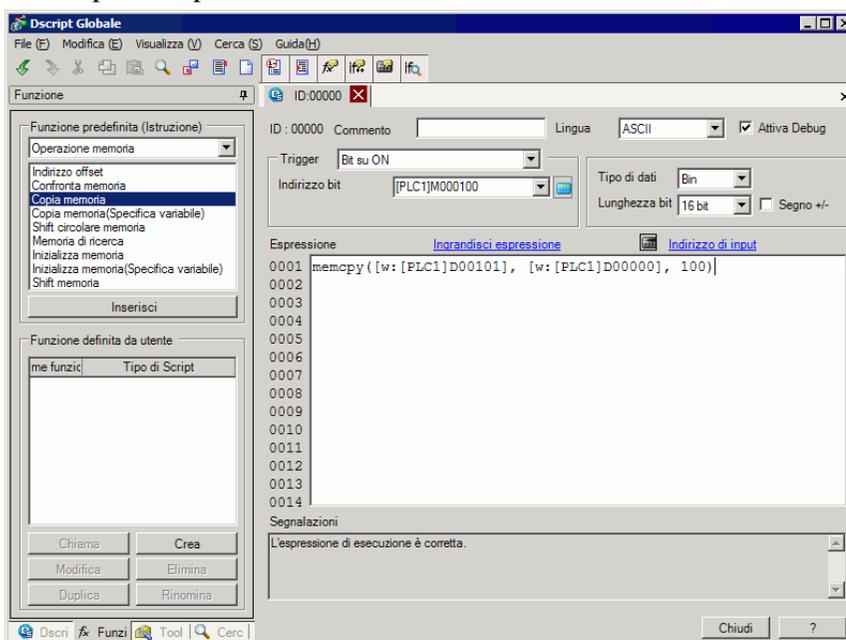


- 8 Per il [Parametro1], inserire D00101, e fare clic su [ENT].



- 9 Per il [Parametro 2] inserire D00000, e fare clic su [OK].

- 10 Lo script è completo.



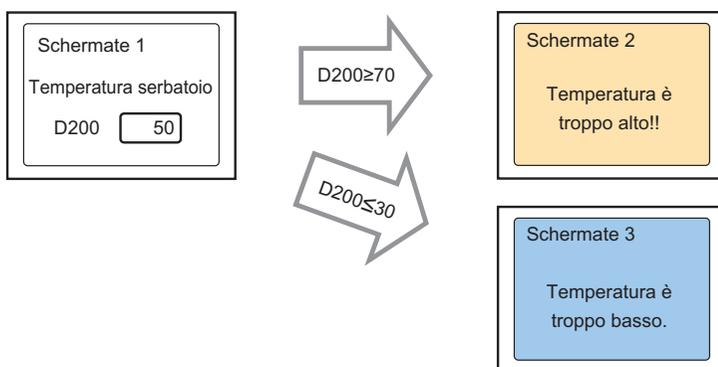
## 21.4 Visualizzare un allarme quando si verifica un errore

**NOTA**

- Fare riferimento alla guida per le impostazioni per altri dettagli.  
☞ "21.9.1 Guida alle impostazioni D-Script/[D-Script globale] comune" (pagina 21-53)
- Vedere quanto segue per altre informazioni sui comandi disponibili per gli script.  
☞ "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)

### Azione

Il sistema di gestione temperatura rileva un bit errore dal dispositivo collegato e visualizza messaggi di allarme quando l'indirizzo di memorizzazione delle informazioni sulla temperatura (D200) arriva a 70°C oppure oltre, o scende al di sotto dei 30°C o a una temperatura inferiore. Inoltre, lo script conterà il numero degli errori rilevati.



L'indirizzo che conta ogni volta D200 si porta a 70° C o a una temperatura inferiore e memorizza il numero di volte LS0300

L'indirizzo che conta ogni volta D200 scende a 30°C o al di sotto di questa temperatura e memorizza il numero di volte LS0301

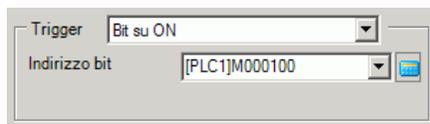
Indirizzo che memorizza il numero di schermo allarme: LS0008

### Comandi utilizzati

Comando	Riepilogo funzioni
if ( )	Quando la condizione "if", inclusa nelle parentesi "( )", è vera, l'espressione che segue l'istruzione "if ( )" verrà eseguita. ☞ "21.11.8 Espressioni condizionali" (pagina 21-147)
Superiore o uguale a (>=)	Vero se N1 è superiore o uguale a N2 (N1 >= N2). ☞ "21.11.9 Confronto" (pagina 21-152)
Assegnazione (=)	Assegnare il valore sulla parte destra alla parte sinistra. ☞ "21.11.10 Operatore" (pagina 21-154)
Aggiunta (+)	Aggiunge una costante ai dati di un dispositivo word. ☞ "21.11.10 Operatore" (pagina 21-154)
Inferiore o uguale a (<=)	Vero se N1 è inferiore o uguale a N2 (N1 <= N2). ☞ "21.11.9 Confronto" (pagina 21-152)

## Trigger

In [Trigger], selezionare [Bit su ON], e impostare l[Indirizzo bit] su M000100.



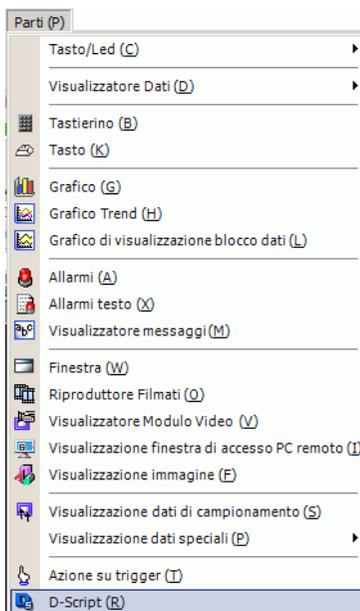
## Script completato

```

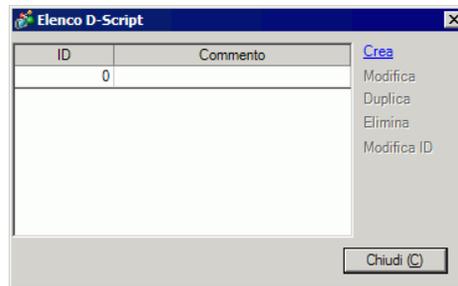
Espressione      Ingrandisci espressione      Inserisci Indirizzo
0001  if ([w:[PLC]D00200]>=70)          //When temperature is 70(or more
0002  {
0003      [w:[#INTERNAL]LS0302]=100      //Substitutes 100, the No. of the70
0004      //Substitutes 100, the No. of the70
0005      [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1 //Increments error count.
0006  }
0007  endif
0008
0009  if ([w:[PLC]D00200]>=30)          //When temperature is 30(or more
0010  {
0011      [w:[#INTERNAL]LS0302]=101      //Substitutes 101, the No. of the70
0012      //Substitutes 101, the No. of the70
0013      [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1 //Increments error count.
0014  }
0015  endif
0016
    
```

## Procedura di creazione

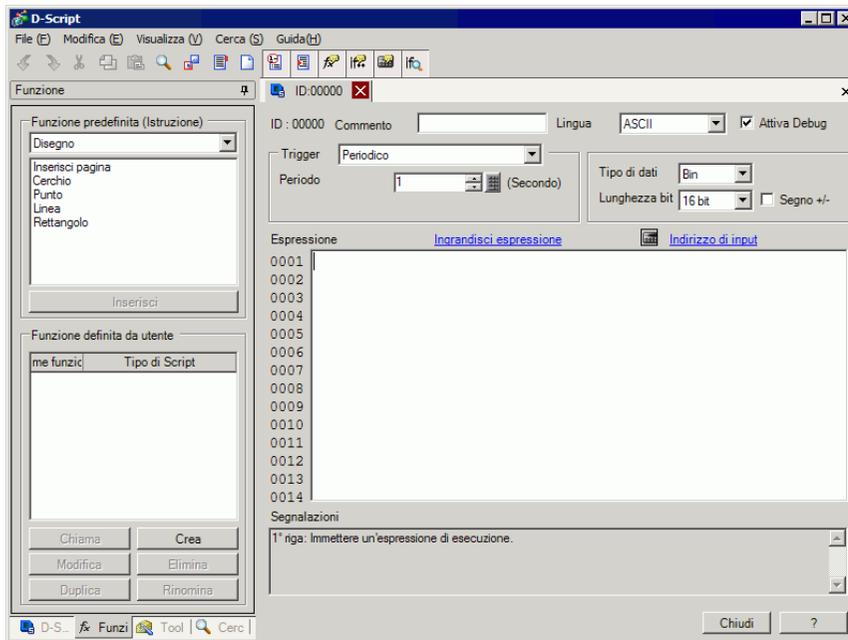
1 Nel menu [Parti (P)], fare clic su [D-Script (R)] o su  .



2 Fare clic su [Creare]. Gli ID per script esistenti sono visualizzati nell'[Elenco D-Script].



3 Viene visualizzata la finestra di dialogo [D-Script].



4 Impostare i commenti. Inserire "Visualizzatore allarme".

5 Selezionare [Bit ON] in [Trigger] e specificare M00100 come [Indirizzo bit].

6 Creare un programma aggiungendo Funzioni, Istruzioni ed Espressioni all'Area Espressioni script per completare lo script.

```

Espressione
0001 if {[w:[PLC]D00200]>=70} //When temperature is 70(or more
0002 {
0003     [w:[#INTERNAL]LS0302]=100 //Substitutes 100, the No. of the70
0004     // (or more warning message screen.
0005     [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1 //Increments error count.
0006 }
0007 endif
0008
0009 if {[w:[PLC]D00200]>=30} //When temperature is 30(or more
0010 {
0011     [w:[#INTERNAL]LS0302]=101 //Substitutes 101, the No. of the70
0012     // (or more warning message screen.
0013     [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1 //Increments error count.
0014 }
0015 endif
0016
0017
    
```

**NOTA**

- Quando si seleziona testo, premere i tasti [CTRL] + [MAIUSC] + [Freccia a destra]/[Freccia a sinistra] per selezionare un intero blocco di testo.
  - Premere il Tasto [CTRL] + [F4] per chiudere il visualizzatore attualmente selezionato.
  - Premere il tasto [Esc] per sovrascrivere lo script o per eliminarlo e uscire.
-

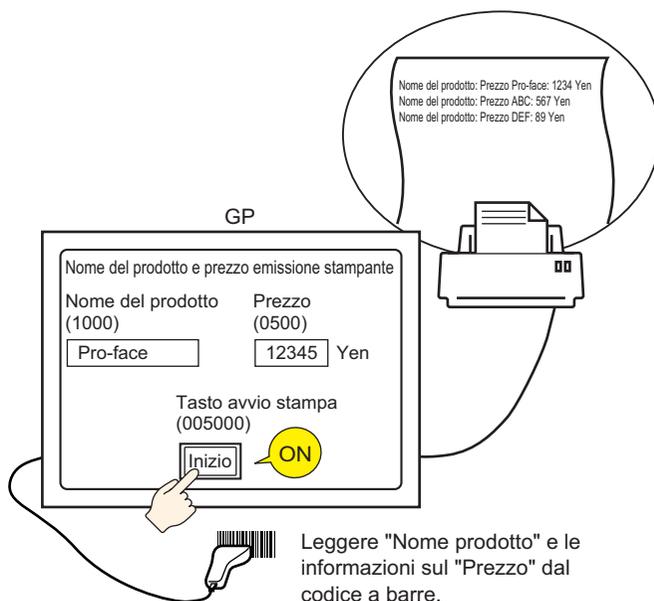
## 21.5 Comunicazione con periferiche non supportate

**NOTA**

- Fare riferimento alla guida per le impostazioni per altri dettagli.  
☞ "21.9.1 Guida alle impostazioni D-Script/[D-Script globale] comune" (pagina 21-53)
- Vedere quanto segue per altre informazioni sui comandi disponibili per gli script.  
☞ "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)

### ■ Operazione

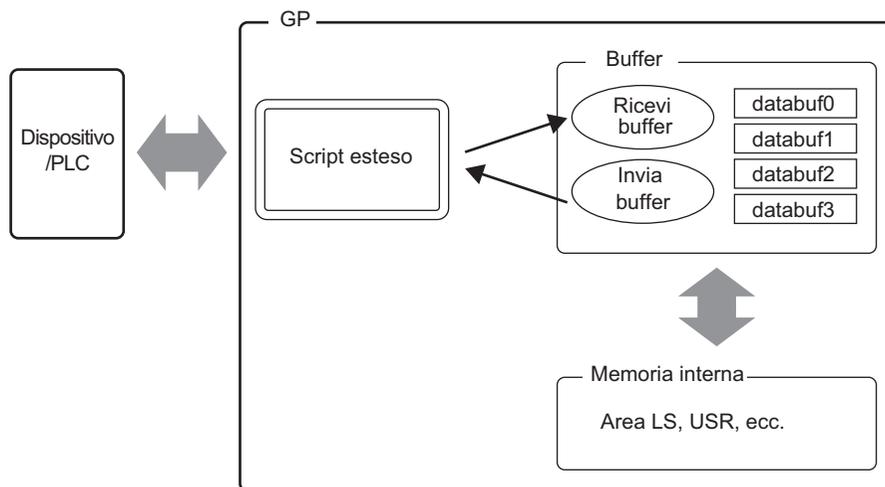
Creare uno script esteso in grado di emettere la lettura dei dati da un lettore di codice a barre collegato all'USB a una stampante seriale collegata a COM 1.



## ■ Struttura degli script estesi

Gli script estesi sono script speciali per comunicare tra la porta seriale interna dell'unità GP e i dispositivi di input/output collegati.

Per la gestione dei dati degli script estesi, come mostrato nella figura seguente, i dati sono memorizzati in databuf0 fino a databuf3 mediante il buffer Invia/Ricevi. Databuf non è diviso da un indirizzo; pertanto, prima di modificare i dati nel dispositivo/PLC, salvarli nella memoria interna.



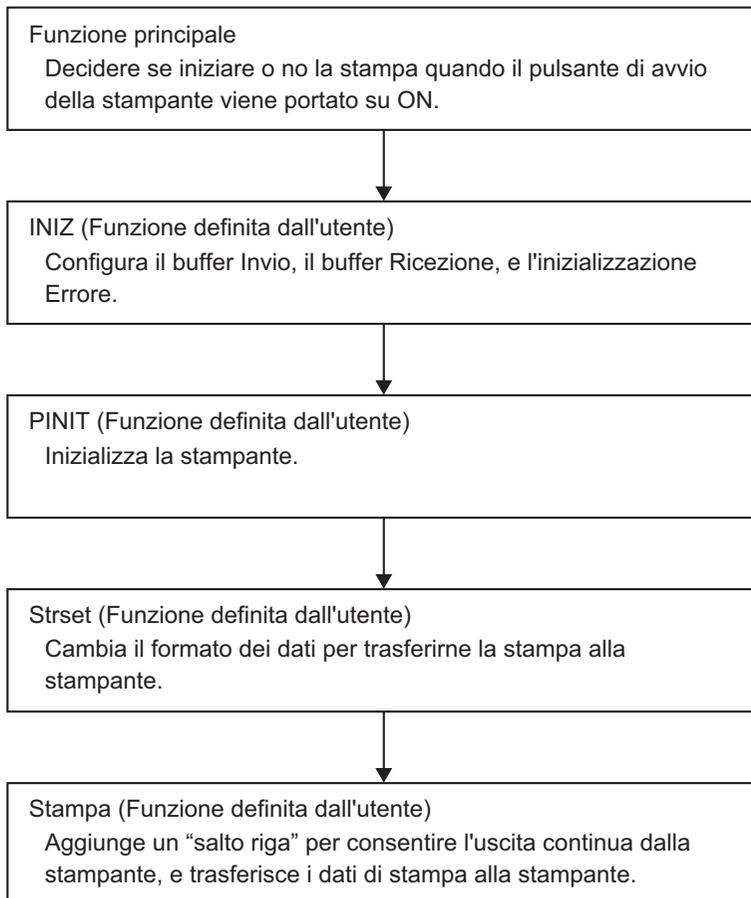
### Ricevi buffer/Invia buffer

Per le comunicazioni con il dispositivo/PLC, questo componente agisce come uno spazio bit di memoria che distingue i dati inviati e ricevuti in tempo reale.

### databuf0-databuf3

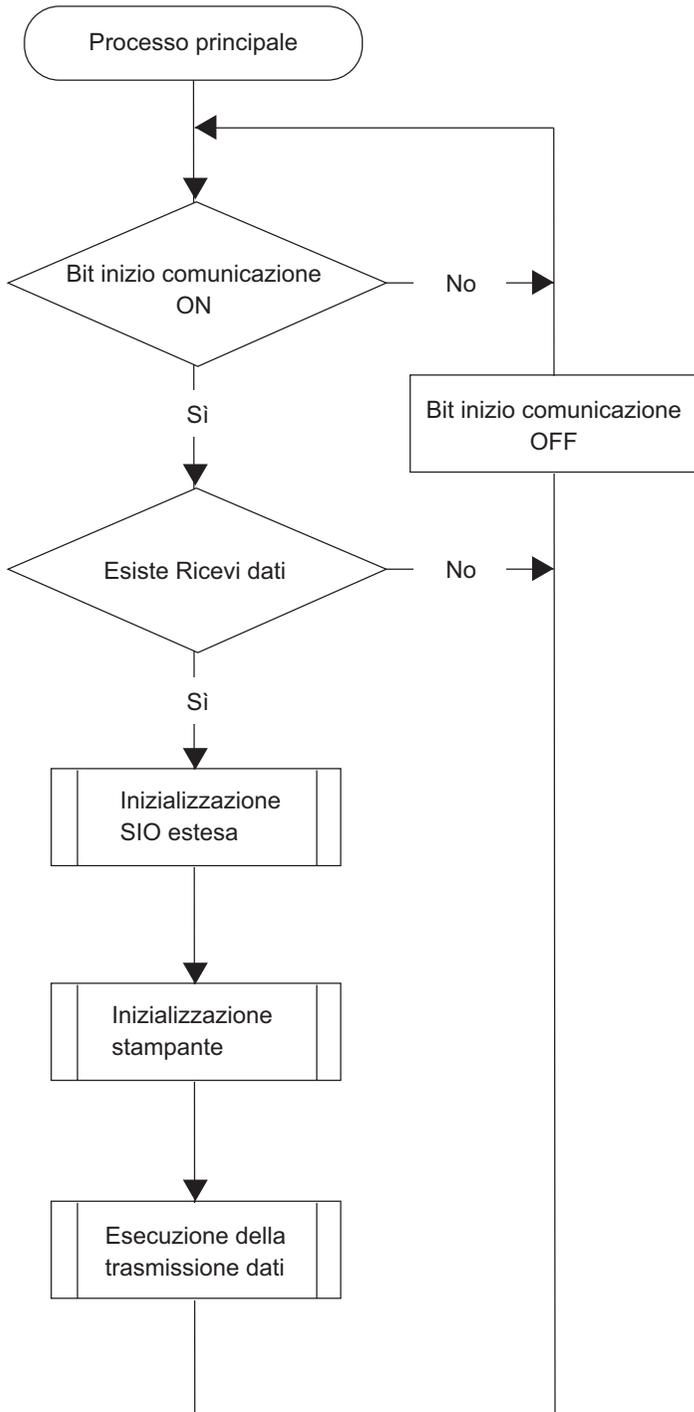
Sono spazi memoria byte (8-bit) usati come punti di memorizzazione dati. Le dimensioni del buffer corrispondono a 1 KB.

## ■ Procedura per creare script

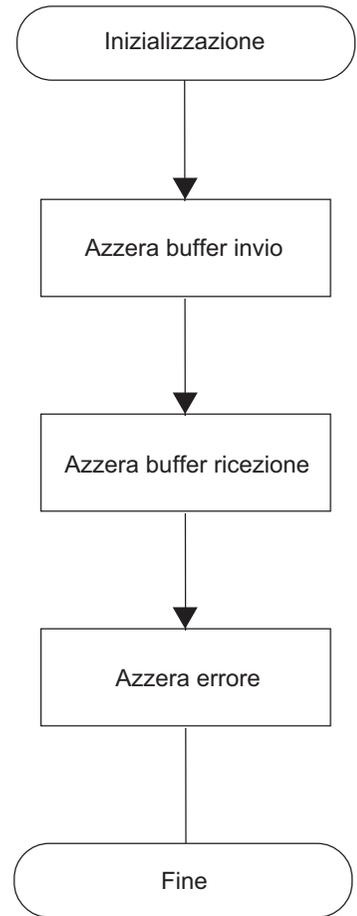


### ■ Diagramma

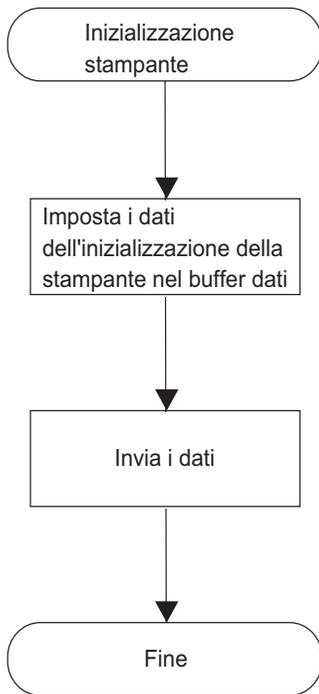
(1) Processo principale



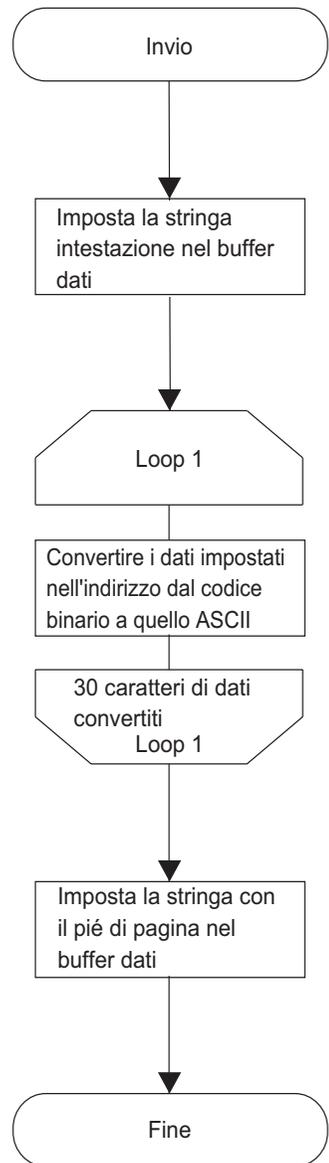
(2) Funzione inizializzazione (INT)



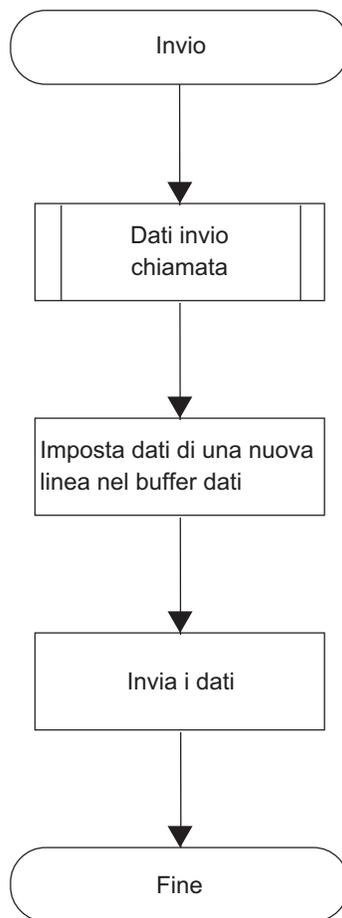
(3) Funzione inizializzazione stampante (PINIT)



(4) Funzione stringa (Strset)



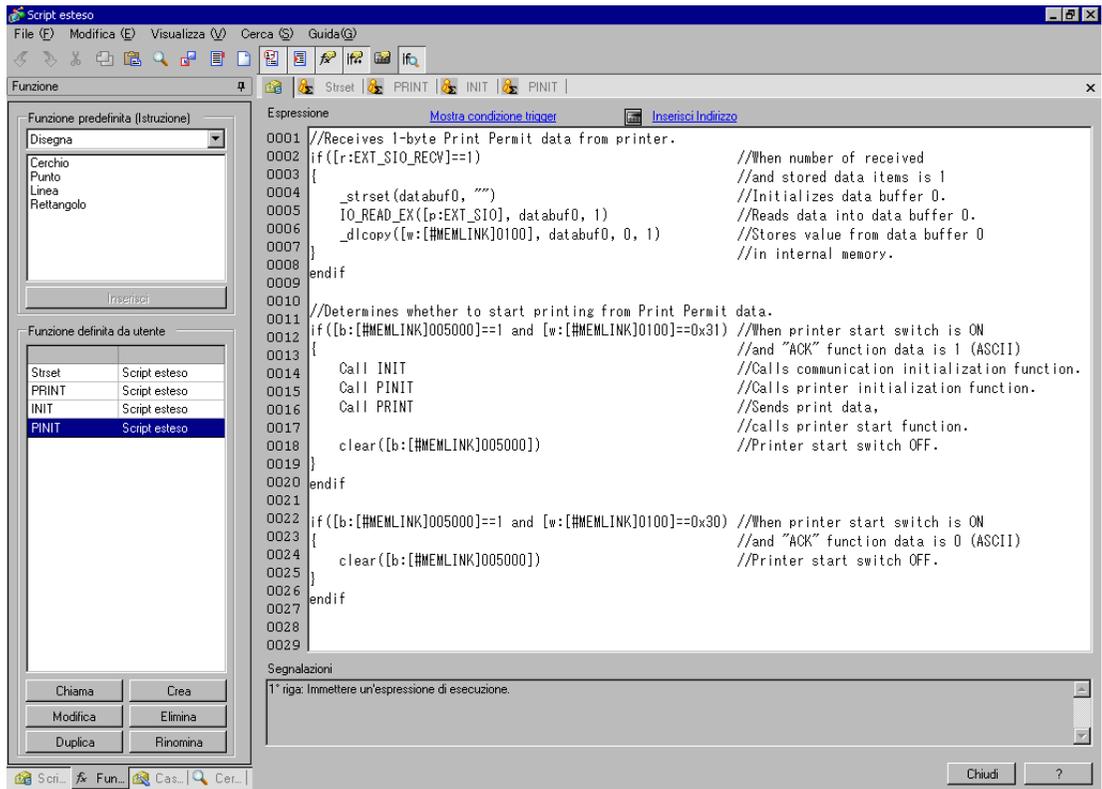
(5) Invia funzione (Stampa)



## ■ Descrizione generale del funzionamento degli script

### ◆ Funzioni principali

Script completato



### Riepilogo funzioni

Quando il pulsante di avvio della stampante (memoria interna 005000) si porta su ON, decide anche se iniziare la stampa dal 1° byte dei dati di Autorizzazione stampa.

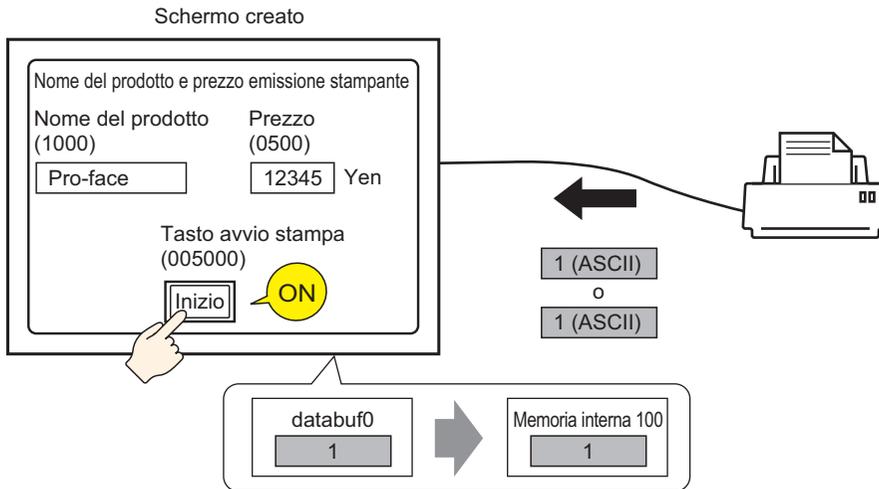
I dati di autorizzazione stampa eseguiranno le seguenti azioni come esempio delle specifiche della stampante.

OK Preparazione stampa: inviare 0x31 (codice ASCII "1") al dispositivo/PLC.

Preparazione stampa non valida: inviare 0x30 (codice ASCII "0") al dispositivo/PLC.

L'unità GP riceve i dati di Autorizzazione stampa in databuf0; questi dati sono spostati in una memoria interna 100 facilmente accessibile con la seguente gestione di script.

Quando la memoria interna è 100 = 0x31 (Codice ASCII per il valore "1"), inizia la stampa.  
 Quando la memoria interna è 0x30 (Codice ASCII per "0") ritorna all'inizio e ripete il processo fino alla ricezione dei dati 0x31.



## ◆ INIT (Funzione definita dall'utente)

Script completato

```

Espressione      Mostra condizione trigger  Inserisci Indirizzo
0001 [c:EXT_SIO_CTRL00]=1 //Clears send buffer.
0002 [c:EXT_SIO_CTRL01]=1 //Clears receive buffer.
0003 [c:EXT_SIO_CTRL02]=1 //Clears error.
0004
    
```

Riepilogo funzioni

Configurare il buffer di invio, il buffer di ricezione e l'inizializzazione errore.

## ◆ PINIT (Funzione definita dall'utente)

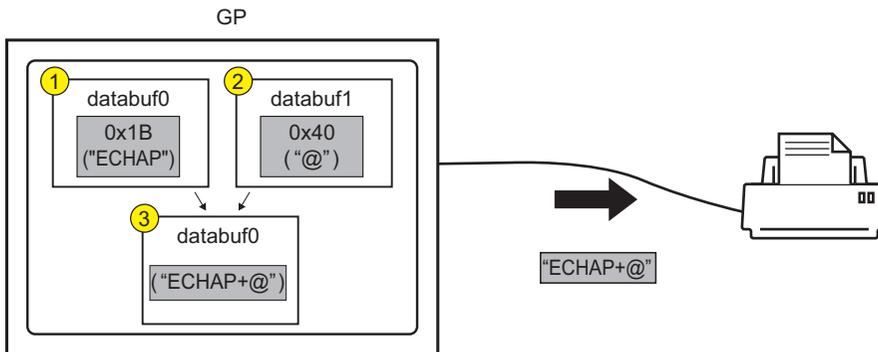
Script completato

```

Espressione      Mostra condizione trigger  Inserisci Indirizzo
0001 //Printer initialization ( ESC/P [ESC + @] )
0002
0003 _strset(databuf0, "") //Clears data buffer 0.
0004 _strset(databuf0, 0x1B) //Sets ASCII code for "ESC".
0005 _strset(databuf1, "") //Clears data buffer 1.
0006 _strset(databuf1, 0x40) //Sets ASCII code for "@".
0007 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0008 _strlen([t:0000], databuf0) //Converts data length to numerical value
0009 //and stores it in temporary address.
0010
0011 //Sends data from serial port.
0012
0013 IO_WRITE_EX([p:EXT_SIO], databuf0, [t:0000]) //Sends amount of buffer 0 data specified by value
0014 //of temporary address.
0015
    
```

## Riepilogo funzioni

Inizializzare la stampante. Inviare il comando ESC/P "ESC+@" alla stampante.



## ◆ Strset (Funzione definita dall'utente)

### Script completato

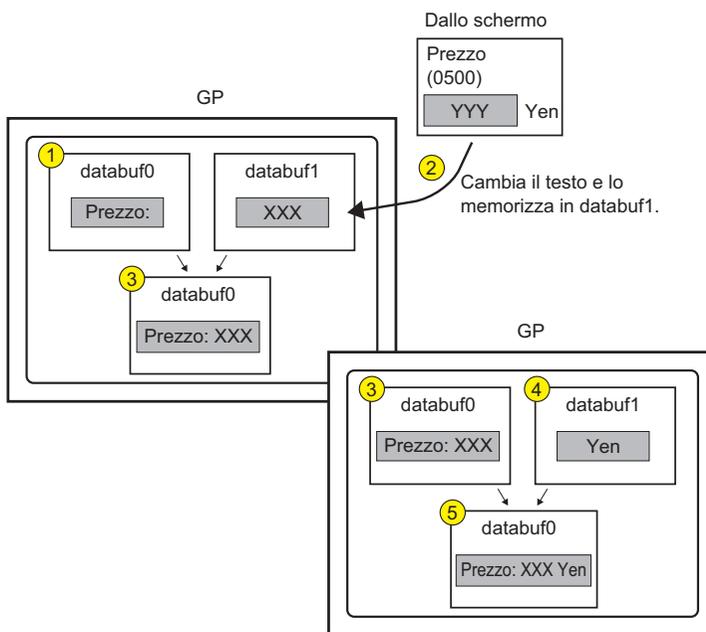
```

Espressione      Mostra condizione trigger      Inserisci Indirizzo
0001 //Appends "Price:" and "Yen" text strings.
0002 _strset(databuf0, "") //Initializes data buffer 0.
0003 _strset(databuf0, "価格 : ") //Price: Stores text string in data buffer 0.
0004 _bin2decase(databuf0, [w:[#MEMLINK]0500]) //Converts numerical value to text string
0005 //and stores it in data buffer 1.
0006 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0007 _strset(databuf1, "") //Initializes data buffer 1.
0008 _strset(databuf1, "円") //Yen Stores text string in data buffer 1.
0009 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0010
0011 //Temporary address initialization
0012 [t:0001]=0
0013 [t:0002]=0
0014
0015 //Re-stores text string stored sequentially in word units in internal memory, in byte units (30 characters of data).
0016 loop()
0017 {
0018     [w:[#MEMLINK]2000]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001] >> 8 //Stores higher-order byte in place of lower-order byte.
0019     [w:[#MEMLINK]2001]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001] & 0xFF //Erases higher-order byte and stores result
0020     //in next address.
0021     [t:0001]=[t:0001]+1 //Address offset + 1
0022     [t:0002]=[t:0002]+2 //Address offset + 2
0023     if([t:0001]=15) //Loop ends when process of storing each 2-byte value
0024     { //in 2 words has repeated 15 times.
0025         break
0026     }
0027     endif
0028 }
0029 endloop
0030 _ldcopy(databuf2, [w:[#MEMLINK]2000], 30) //Stores data in internal memory addresses 2000 to 2030 in data buffer
0031 //as text string.
0032 //Appends "Product Name:" text string.
0033 _strset(databuf1, "") //Initializes data buffer 1.
0034 _strset(databuf1, "品名 : ") //Product Name: Stores text string in data buffer 1.
0035 _strcat(databuf1, databuf2) //Concatenates data buffer 2 to end of data buffer 1.
0036
0037 //Appends price character string to product name.
0038 _strcat(databuf1, databuf0) //Concatenates value of data buffer 0 to data buffer 1.
0039

```

Riepilogo funzioni

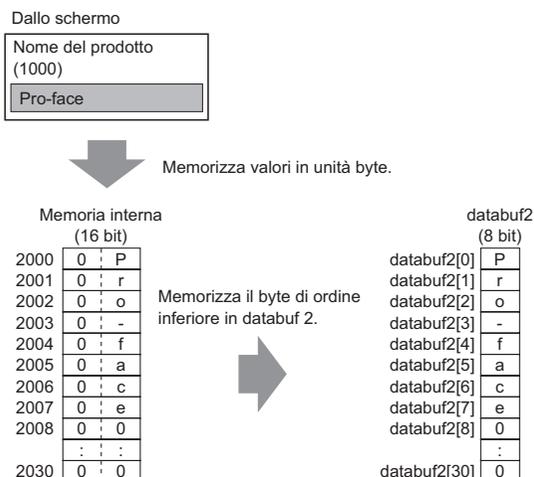
1 Apporre il testo “Prezzo:” e “Yen” ai dati prezzo salvati nella memoria interna 0500.



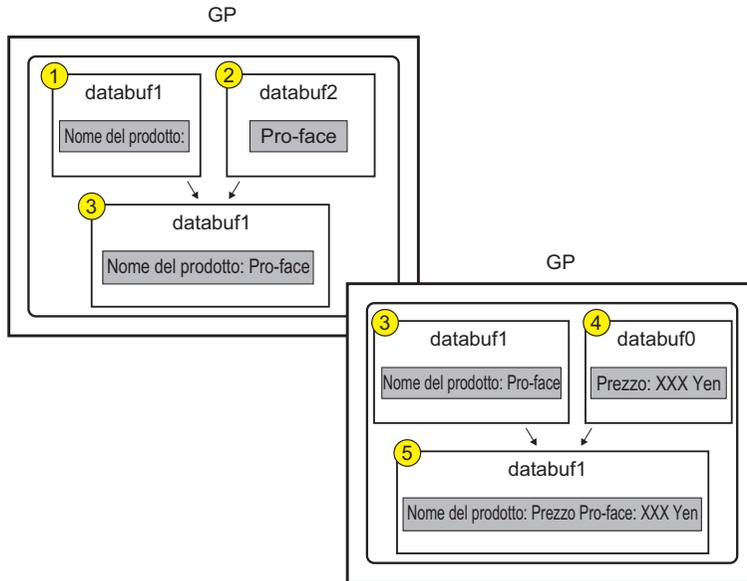
2 Cambiare il formato dei dati in modo da inviare dati stampa alla stampante. Dividere i dati stringa (Nome prodotto) memorizzati in sequenza nella memoria interna 1000 in unità byte, e memorizzare nella memoria interna da 2000 a 2030 come dati stringa byte di ordine basso. Usare la funzione `_ldcopy` e memorizzare i dati in databuf2 nell'ordine del byte più basso dell'indirizzo word consecutivo.

**NOTA**

- La funzione `_ldcopy` prende i dati memorizzati come word, e salva solo i byte degli ordini inferiori nel buffer, mentre i dati dei byte degli ordini superiori vengono ignorati.



3 Allegare il testo "Nome prodotto:" e "Prezzo" a databuf2.



◆ **Stampa (Funzione definita dall'utente)**

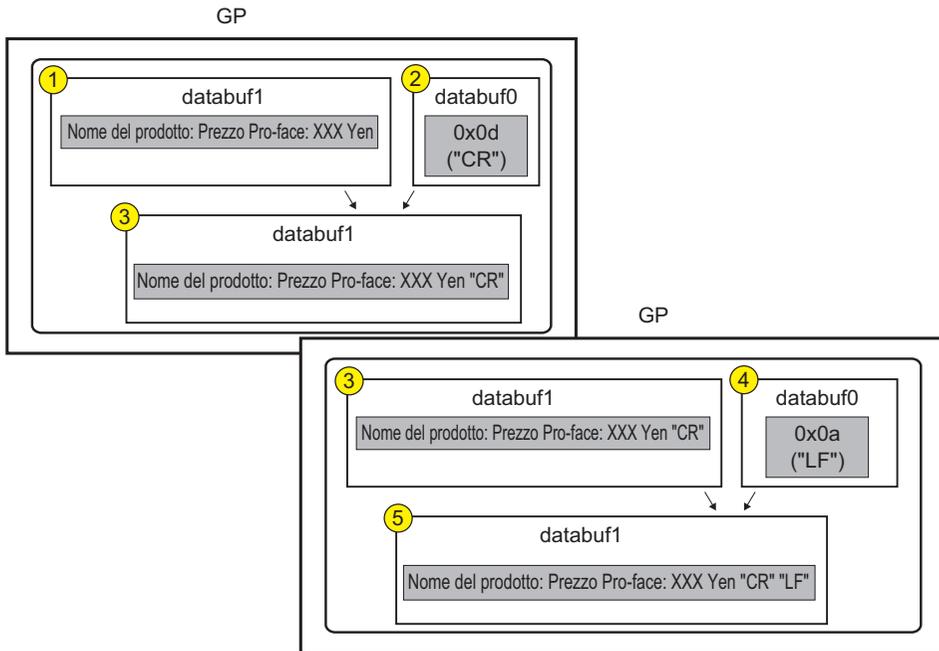
Script completato

```

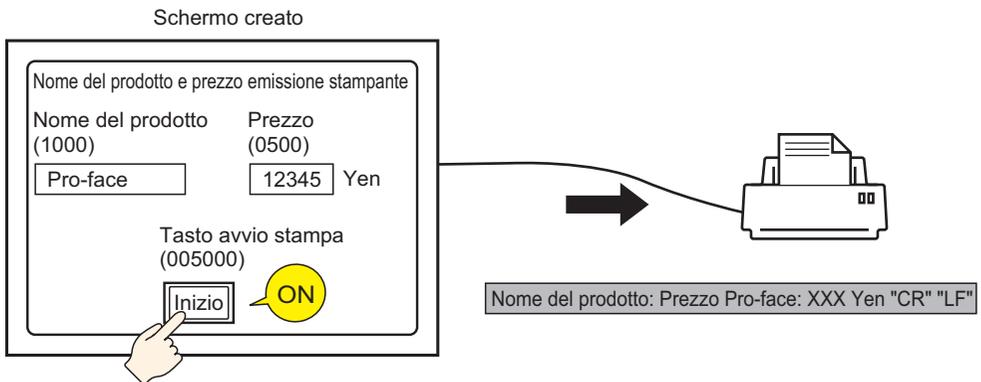
Espressione      Mostra condizione trigger  Inserisci indirizzo
0001 Call Strset //Calls print data function.
0002 _strset(databuf0, "") //Clears data buffer 1.
0003
0004 //Printing and delimiter (line feed + carriage return)
0005
0006 _strset(databuf0, 0x0d) //Prints and returns to head of line.
0007 _strcat(databuf1, databuf0) //Concatenates data buffer 1 to end of data buffer 0.
0008 _strset(databuf0, "") //Clears data buffer 1.
0009 _strset(databuf0, 0x0a) //Sends line feed to move to next line.
0010 _strcat(databuf1, databuf0) //Concatenates data buffer 1 to end of data buffer 0.
0011
0012 _strlen([t:0000], databuf1) //Converts data length to numerical value
0013 //and stores it in temporary address.
0014 //Sends data from serial port.
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000]) //Sends amount of buffer 0 data
0017 //specified by value of temporary address.
0018
    
```

## Riepilogo funzioni

1 Apporre un "salto riga" per l'emissione stampante continua.



2 Inviare i dati di stampa alla stampante.

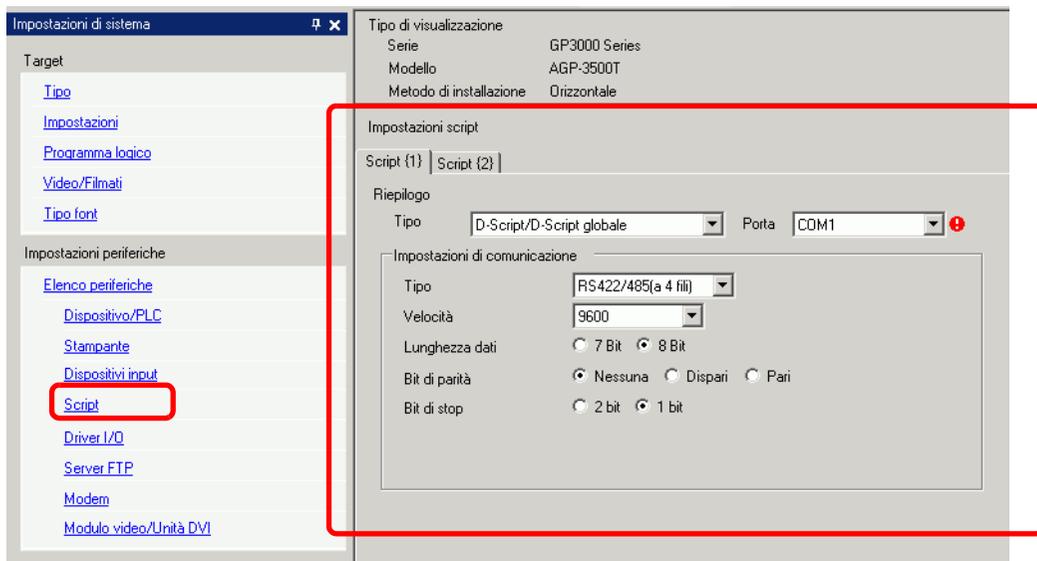


## ■ Comandi utilizzati

Comando	Riepilogo funzioni
if ( )	Quando la condizione "if", inclusa nelle parentesi "( )", è vera, l'espressione che segue l'istruzione "if ( )" verrà eseguita. ☞ "21.11.8 Espressioni condizionali" (pagina 21-147)
Impostazioni etichette [r:EXT_SIO_RECV]	Mostra la quantità di dati (numero di byte) ricevuta in quel momento. Le dimensioni dei dati ricevuti costituiscono una funzione di sola lettura. ☞ "21.11.4 Operazione porta SIO" (pagina 21-98)
Equivalente (==)	Vero se N1 equivale a N2 (N1 = N2). ☞ "21.11.9 Confronto" (pagina 21-152)
Impostazioni testo (_strset)	Una stringa fissa viene memorizzata nel buffer dati. ☞ "21.11.11 Operazione del testo" (pagina 21-158)
Ricevi esteso (IO_READ_EX)	Riceve dati delle dimensioni indicate in Dimensioni dati ricevuti (byte) dal SIO esteso e li memorizza nel buffer dati. ☞ "21.11.4 Operazione porta SIO" (pagina 21-98)
Dal buffer dati al dispositivo interno (_dlcopy)	Ciascun byte dei dati della stringa memorizzato nell'offset del buffer dati viene copiato nell'area LS in base al numero di stringhe. ☞ "21.11.11 Operazione del testo" (pagina 21-158)
Impostazioni etichette [c:EXT_SIO_CTRL**]	La variabile Controllo viene utilizzata per azzerare il buffer Invia, il buffer Ricevi e lo stato di errore. ☞ "21.11.8 Espressioni condizionali" (pagina 21-147)
Connetti testo (_strcat)	Una stringa carattere o un codice carattere sono concatenati con il buffer di testo. ☞ "21.11.11 Operazione del testo" (pagina 21-158)
Lunghezza testo (_strlen)	Ottiene la lunghezza della stringa memorizzata. ☞ "21.11.11 Operazione del testo" (pagina 21-158)
Invia esteso (IO_WRITE_EX)	Invia i dati nel buffer dati con il SIO esteso secondo le dimensioni del numero di byte Invia. ☞ "21.11.4 Operazione porta SIO" (pagina 21-98)
Assegnazione (=)	Assegnare il valore sulla parte destra alla parte sinistra. ☞ "21.11.10 Operatore" (pagina 21-154)
Aggiunta (+)	Aggiunge una costante ai dati di un dispositivo word. ☞ "21.11.10 Operatore" (pagina 21-154)
Conversione del valore numerico in una stringa decimale (_bin2decasc)	Questa funzione viene utilizzata per convertire un numero intero in una stringa decimale. ☞ "21.11.11 Operazione del testo" (pagina 21-158)
Dal dispositivo interno al buffer dati (_ldcopy)	I dati della stringa memorizzata nell'area LS area vengono copiati nel buffer dati secondo il numero di stringhe in un trasferimento byte-per-byte. ☞ "21.11.11 Operazione del testo" (pagina 21-158)

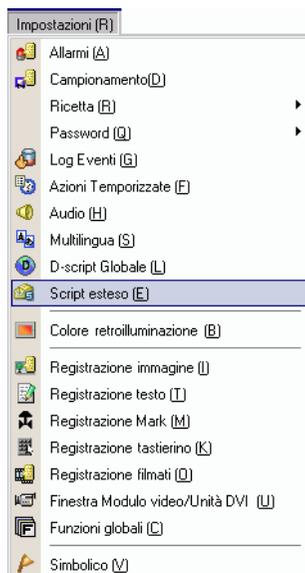
## ■ Procedura di creazione

- 1 Configurare le impostazioni dello script in modo da usare Script esteso per comunicare. Nel menu [Progetto (F)], fare clic su [Impostazioni di sistema (C)]. Selezionare [Script]. Impostare il [Tipo] di [Script esteso].



Esistono due etichette per le impostazioni di script. Impostare [Porta] su COM1 o COM2. Configurare le [Impostazioni di comunicazione] in modo da farle corrispondere al SIO esteso.

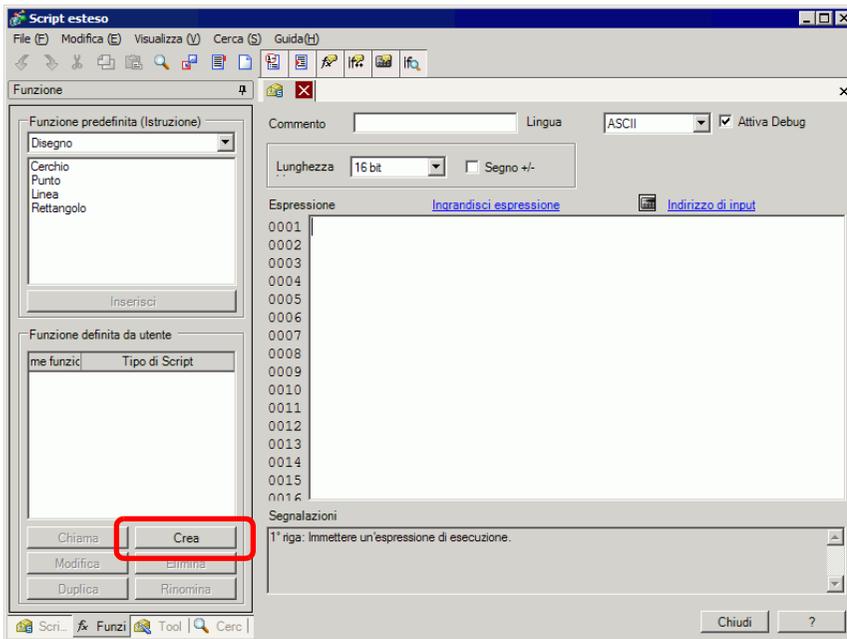
- 2 Nel menu [Impostazioni comuni (R)], selezionare [Impostazioni Script estese (E)].



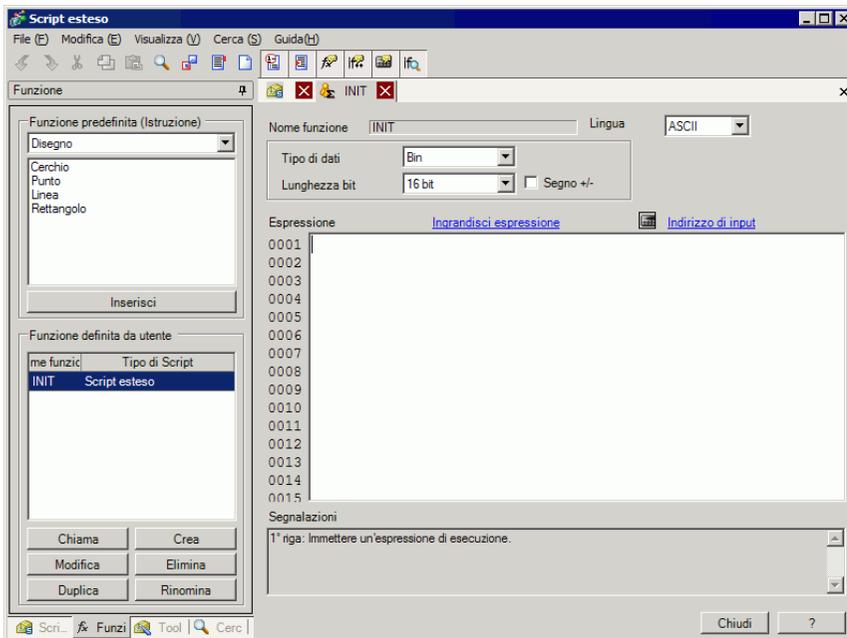
### NOTA

- Quando appare il messaggio "Verrà usato lo script esteso. Continuare?", fare clic su [Sì].

3 Registra "INIT" come funzione definita dall'utente. Fare clic sulla scheda [Funzione] e fare clic sul pulsante [Crea] del frame della funzione definita dall'utente.



4 Inserire [INIT] come il nome della funzione, fare clic su [OK]. Apparirà lo schermo seguente.



5 Creare uno script nell'Espressione esecuzione con comandi, istruzioni e inserimento delle costanti.

```

Espressione
Mostra condizione trigger
Inserisci Indirizzo
0001 [c:EXT_SIO_CTRL00]=1 //Clears send buffer.
0002 [c:EXT_SIO_CTRL01]=1 //Clears receive buffer.
0003 [c:EXT_SIO_CTRL02]=1 //Clears error.
0004
    
```

6 Allo stesso modo, registrare "PINIT" come una funzione definita dall'utente. Inserire [PINIT] come nome della funzione e creare il seguente script in Espressione esecuzione.

```

Espressione
Mostra condizione trigger
Inserisci Indirizzo
0001 //Printer initialization ( ESC/P [ESC + @] )
0002
0003 _strset(databuf0, "") //Clears data buffer 0.
0004 _strset(databuf0, 0x1B) //Sets ASCII code for "ESC".
0005 _strset(databuf1, "") //Clears data buffer 1.
0006 _strset(databuf1, 0x40) //Sets ASCII code for "@".
0007 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0008 _strlen([t:0000], databuf0) //Converts data length to numerical value
0009 //and stores it in temporary address.
0010
0011 //Sends data from serial port.
0012
0013 IO_WRITE_EX([p:EXT_SIO], databuf0, [t:0000]) //Sends amount of buffer 0 data specified by value
0014 //of temporary address.
0015
    
```

7 Allo stesso modo, registrare "Strset" come una funzione definita dall'utente. Inserire [Strset] come nome della funzione e creare il seguente script in Espressione esecuzione.

```

Espressione
Mostra condizione trigger
Inserisci Indirizzo
0001 //Appends "Price:" and "Yen" text strings.
0002 _strset(databuf0, "") //Initializes data buffer 0.
0003 _strset(databuf0, "価格 : ") //Price: Stores text string in data buffer 0.
0004 _bin2decasc(databuf0, [w:[#MEMLINK]0500]) //Converts numerical value to text string
0005 //and stores it in data buffer 1.
0006 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0007 _strset(databuf1, "") //Initializes data buffer 1.
0008 _strset(databuf1, "円") //Yen Stores text string in data buffer 1.
0009 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0010
0011 //Temporary address initialization
0012 [t:0001]=0
0013 [t:0002]=0
0014
0015 //Re-stores text string stored sequentially in word units in internal memory, in byte units (30 characters of data).
0016 loop()
0017 {
0018     [w:[#MEMLINK]2000]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001] >> 8 //Stores higher-order byte in place of lower-order byte.
0019     [w:[#MEMLINK]2001]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001] & 0xFF //Erases higher-order byte and stores result
0020     //in next address.
0021     [t:0001]=[t:0001]+1 //Address offset + 1
0022     [t:0002]=[t:0002]+2 //Address offset + 2
0023     if ([t:0001]==15) //Loop ends when process of storing each 2-byte value
0024     { //in 2 words has repeated 15 times.
0025         break
0026     }
0027 }
0028 }
0029 endloop
0030 _ldcopy(databuf2, [w:[#MEMLINK]2000], 30) //Stores data in internal memory addresses 2000 to 2030 in data buffer
0031 //as text string.
0032 //Appends "Product Name:" text string.
0033 _strset(databuf1, "") //Initializes data buffer 1.
0034 _strset(databuf1, "品名 : ") //Product Name: Stores text string in data buffer 1.
0035 _strcat(databuf1, databuf2) //Concatenates data buffer 2 to end of data buffer 1.
0036
0037 //Appends price character string to product name.
0038 _strcat(databuf1, databuf0) //Concatenates value of data buffer 0 to data buffer 1.
0039
    
```

8 Allo stesso modo, registrare "Stampa" come una funzione definita dall'utente. Inserire [PINIT] come nome della funzione e creare il seguente script in Espressione esecuzione.

```

Espressione          Mostra condizione trigger          Inserisci Indirizzo
0001 Call Strset          //Calls print data function.
0002 _strset(databuf0, "") //Clears data buffer 1.
0003
0004 //Printing and delimiter (line feed + carriage return)
0005
0006 _strset(databuf0, 0x0d) //Prints and returns to head of line.
0007 _strcat(databuf1, databuf0) //Concatenates data buffer 1 to end of data buffer 0.
0008 _strset(databuf0, "") //Clears data buffer 1.
0009 _strset(databuf0, 0x0a) //Sends line feed to move to next line.
0010 _strcat(databuf1, databuf0) //Concatenates data buffer 1 to end of data buffer 0.
0011
0012 _strlen([t:0000], databuf1) //Converts data length to numerical value
0013 //and stores it in temporary address.
0014 //Sends data from serial port.
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000]) //Sends amount of buffer 0 data
0017 //specified by value of temporary address.
0018

```

9 Creare lo script principale. Creare lo script seguente in Espressione esecuzione per completare lo script.

```

Espressione          Mostra condizione trigger          Inserisci Indirizzo
0001 //Receives 1-byte Print Permit data from printer.
0002 if([r:EXT_SIO_RECV]==1) //When number of received
0003 { //and stored data items is 1
0004   _strset(databuf0, "") //Initializes data buffer 0.
0005   IO_READ_EX([p:EXT_SIO], databuf0, 1) //Reads data into data buffer 0.
0006   _dlcopy([w:[#MEMLINK]0100], databuf0, 0, 1) //Stores value from data buffer 0
0007 } //in internal memory.
0008 endif
0009
0010 //Determines whether to start printing from Print Permit data.
0011 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0x31) //When printer start switch is ON
0012 { //and "ACK" function data is 1 (ASCII)
0013   Call INIT //Calls communication initialization function.
0014   Call PINIT //Calls printer initialization function.
0015   Call PRINT //Sends print data,
0016 //calls printer start function.
0017   clear([b:[#MEMLINK]005000]) //Printer start switch OFF.
0018 }
0019 endif
0020
0021 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0x30) //When printer start switch is ON
0022 { //and "ACK" function data is 0 (ASCII)
0023   clear([b:[#MEMLINK]005000]) //Printer start switch OFF.
0024 }
0025 endif

```

#### NOTA

- Al momento di collocare le funzioni definite dall'utente create nei passaggi da 3 a 9 nello script principale, selezionare la funzione da collocare e fare clic su [Richiama] nella scheda [Funzione]. La funzione sarà collocata utilizzando "Richiama nome funzione".
- Quando si seleziona testo, premere i tasti [CTRL] + [MAIUSC] + [Freccia a destra]/[Freccia a sinistra] per selezionare un intero blocco di testo.
- Premere il Tasto [CTRL] + [F4] per chiudere il visualizzatore attualmente selezionato.
- Premere il tasto [Esc] per sovrascrivere lo script o per eliminarlo e uscire.

## 21.6 Riferimento ad altri script

### 21.6.1 Introduzione

È possibile visualizzare una funzione definita dall'utente a fianco di un D-Script, un D-script globale, uno Script esteso o di un'altra funzione definita dall'utente.

È possibile scrivere la funzione e nello stesso tempo confrontarla con l'altra funzione, oppure modificare contemporaneamente le due funzioni.

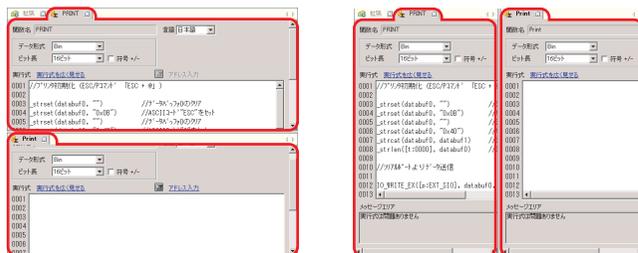
Fare clic sulle schede si passa da uno schermo all'altro.



Visualizzare/modificare con due schermi affiancati

Affiancare in orizzontale

Affiancare in orizzontale



**NOTA**

- È inoltre possibile modificare contemporaneamente script di tipi diversi definiti dall'utente.

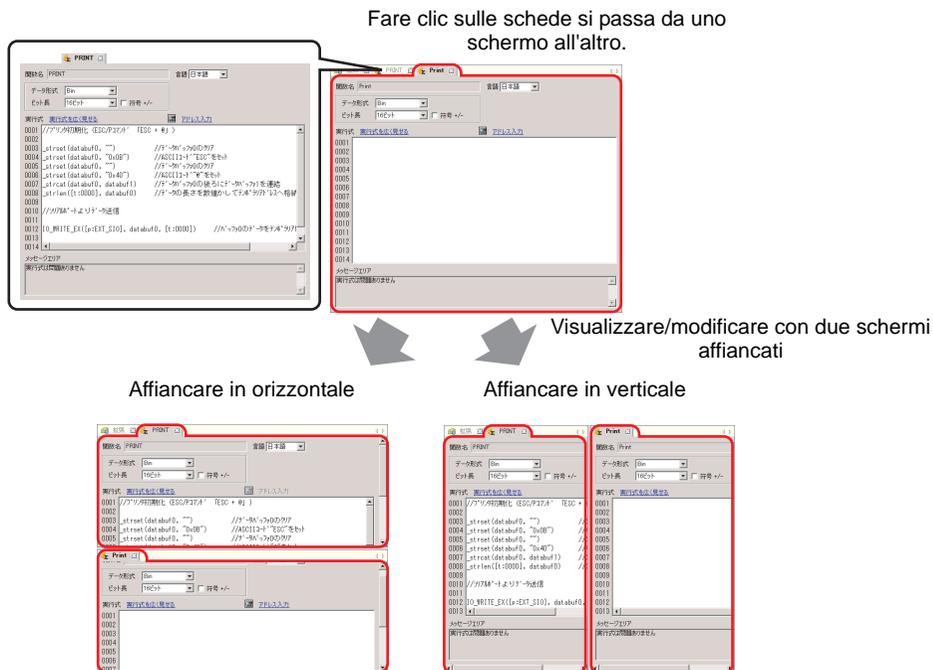
## 21.6.2 Procedura di funzionamento

**NOTA**

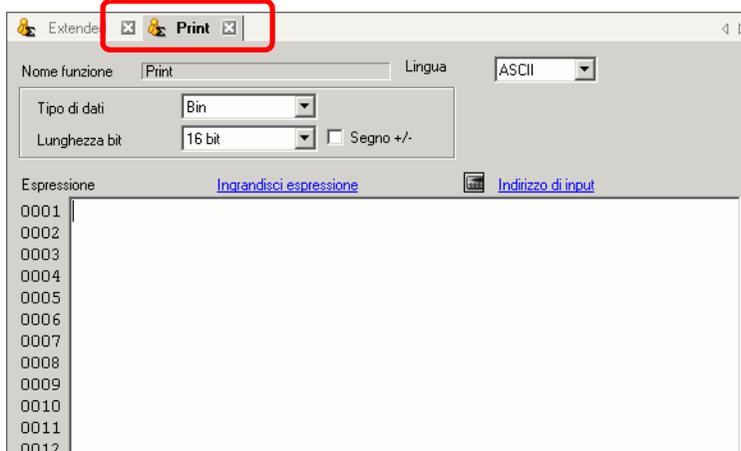
- Fare riferimento alla guida per le impostazioni per altri dettagli.
- ☞ "21.9.1 Guida alle impostazioni D-Script/[D-Script globale] comune" (pagina 21-53)
- Vedere quanto segue per altre informazioni sui comandi disponibili per gli script.
- ☞ "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)
- Vedere quanto segue per altre informazioni sui comandi disponibili per gli script.
- ☞ "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)

Nell finestra di dialogo [D-Script], dividere lo schermo in due schermi in senso orizzontale o verticale.

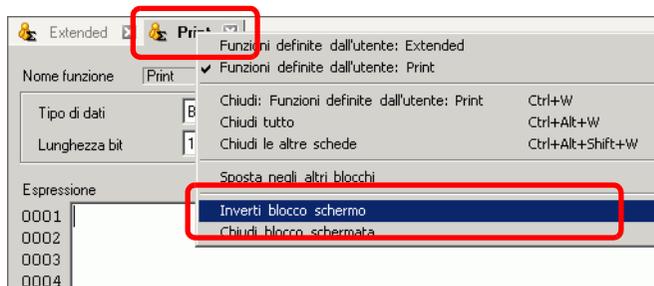
Ad esempio, mentre si visualizza la funzione "PRINT" creata dall'utente in una procedura precedente, si può creare "Stampa".



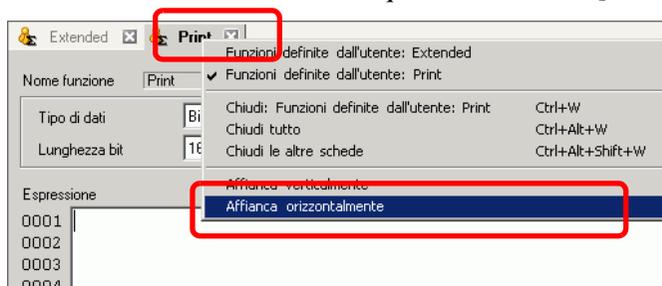
- 1 Nella finestra di dialogo [D-Script], aprire lo script e la funzione definita dall'utente da visualizzare contemporaneamente.



- 2 Se si desidera affiancarli orizzontalmente, fare clic con il pulsante destro del mouse sulla "scheda schermo da visualizzare in basso", quindi fare clic su [Affianca in orizzontale].

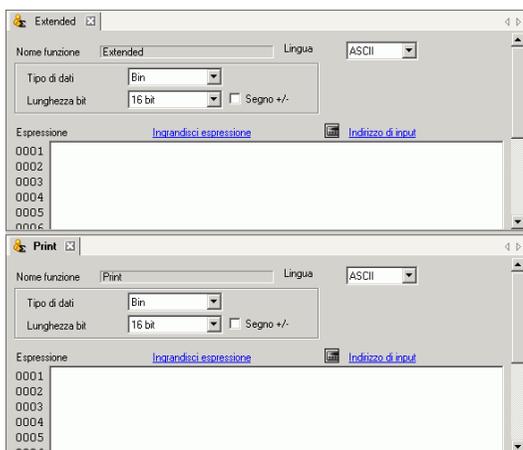


- Se si desidera affiancarli verticalmente, fare clic con il pulsante destro del mouse sulla "scheda schermo da visualizzare a destra", quindi fare clic su [Affianca in verticale].

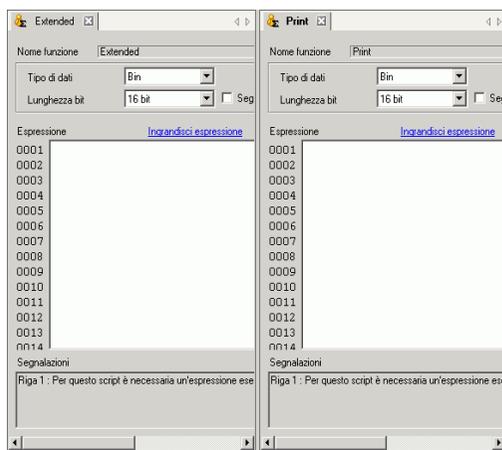


### 3 Gli schermi sono affiancati in orizzontale o in verticale.

Affiancare in orizzontale



Affiancare in verticale



## 21.7 Creare script

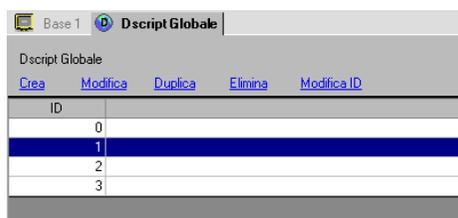
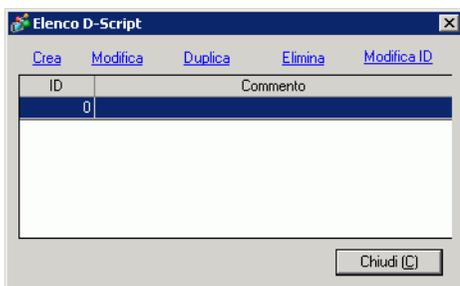
### 21.7.1 Procedura per creare D-Script/D-Script globali

Nel menu [Parti (P)], selezionare [D-Script (R)].

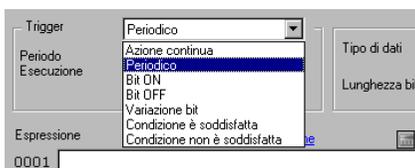
Nel menu [Impostazioni comuni (R)], selezionare [Impostazioni D-Script Globale (L)].

Fare clic su [Creare]. Per vedere uno script esistente, selezionare il numero di ID e fare clic su [Modifica], oppure fare doppio clic sulla riga del numero dell'ID.

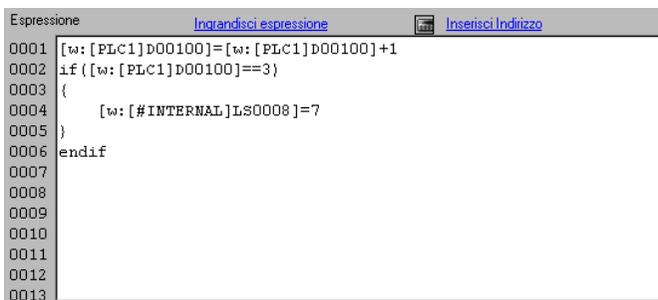
Fare clic su [Creare]. Per vedere uno script esistente, selezionare il numero di ID e fare clic su [Modifica], oppure fare doppio clic sulla riga del numero dell'ID.



Impostare la condizione di trigger che causerà l'esecuzione dello script. Per altre informazioni su questa funzione, fare riferimento a "21.8 Configurazione condizione trigger" (pagina 21-46).



Creare lo script (Espressione esecuzione). Per ulteriori informazioni sui comandi e le funzioni, fare riferimento a "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69).

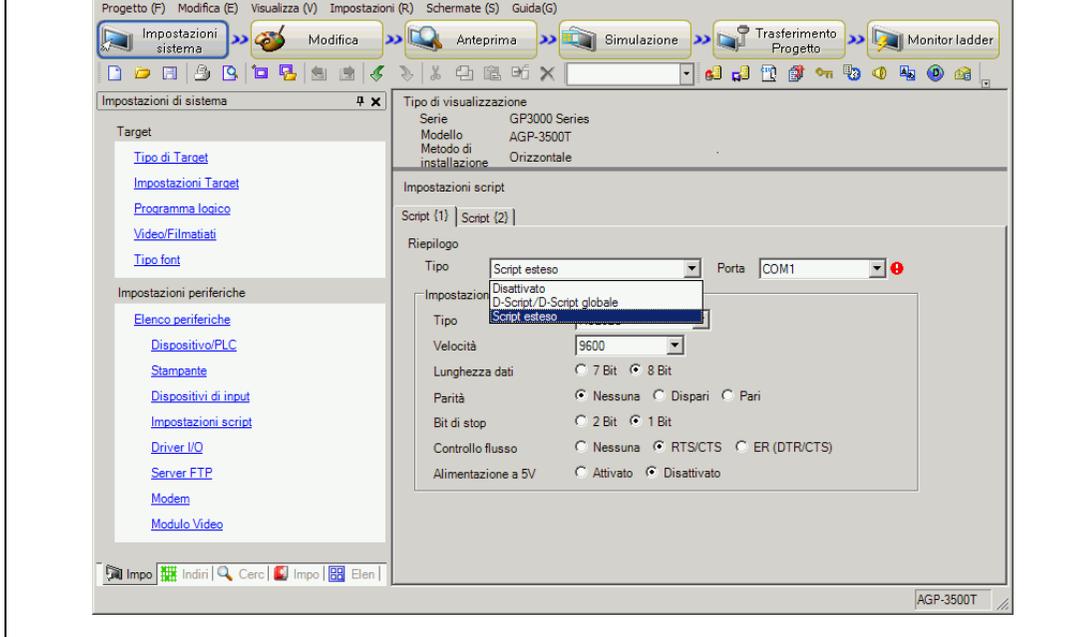


**NOTA**

- La barra componenti visualizza i D-Script registrati nell'ordine in cui sono stati creati. Per cambiare l'ordine nella barra componenti, cambiare il numero di ID per le parti registrate, quindi nel menu [Modifica] selezionare [Barre allineamento automatico]. Si possono modificare le impostazioni dell'ID facendo doppio clic sugli elementi nella barra componenti per visualizzare la finestra di dialogo Modifica.
-

## 21.7.2 Procedura per creare script estesi

Nel menu [Progetto (F)], selezionare [Impostazioni di sistema (C)]. Fare clic su [Impostazioni di I/O script] per visualizzare la finestra di dialogo seguente. Quando si utilizza uno script esteso, selezionare [Tipo] di [Script esteso] e selezionare una [Porta].



Nel menu [Impostazioni comuni (R)], selezionare [Impostazioni Script estese (E)].



Creare lo script (Espressione esecuzione). Per ulteriori informazioni sui comandi e le funzioni, fare riferimento a "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69).

```

Espressione
Ingrandisci espressione
Inserisci Indirizzo
0001 [w: [PLC1]D00100]=[w: [PLC1]D00100]+1
0002 if { [w: [PLC1]D00100]==3}
0003 {
0004     [w: [#INTERNAL]L80008]=7
0005 }
0006 endif
0007
0008
0009
0010
0011
0012
0013
    
```

### 21.7.3 Impostazioni di funzioni definite dall'utente

Registrare uno script esistente come funzione definita dall'utente e che potrà essere usato entro altri script. La funzione registrata può essere utilizzata da un D-Script, un D-Script Globale o da uno Script Esteso.

#### ■ Procedura di impostazione

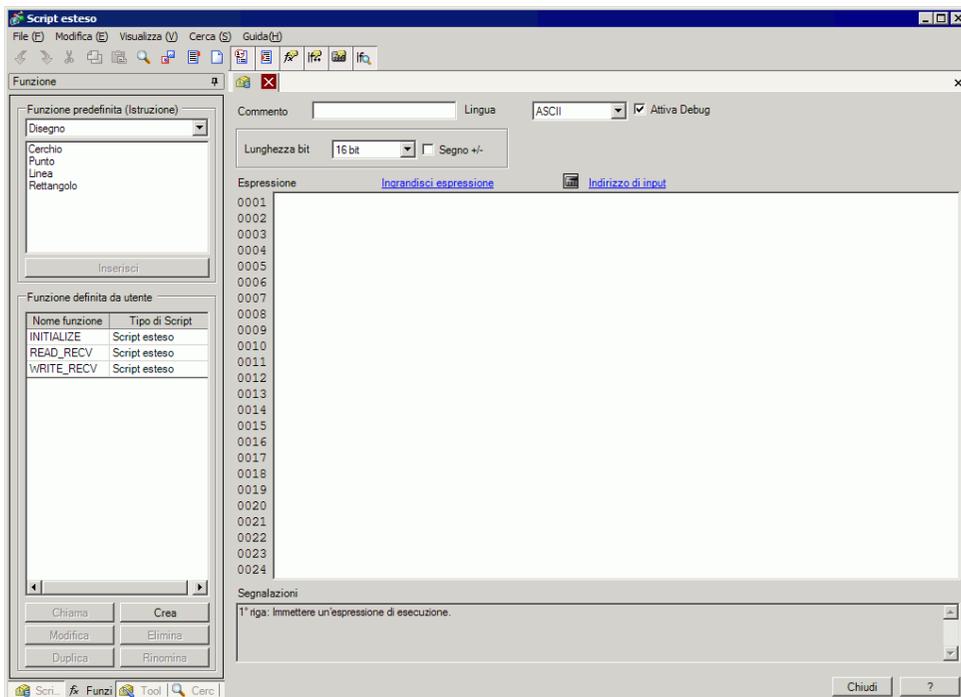
Quando si crea una nuova funzione definita dall'utente  
Fare clic su [Crea]. Apparirà la funzione registrata definita dall'utente.

Quando modificare una funzione definita dall'utente precedentemente registrata

Selezionare la Funzione definita dall'utente che si desidera modificare e fare clic su [Modifica]. Apparirà la funzione registrata definita dall'utente.



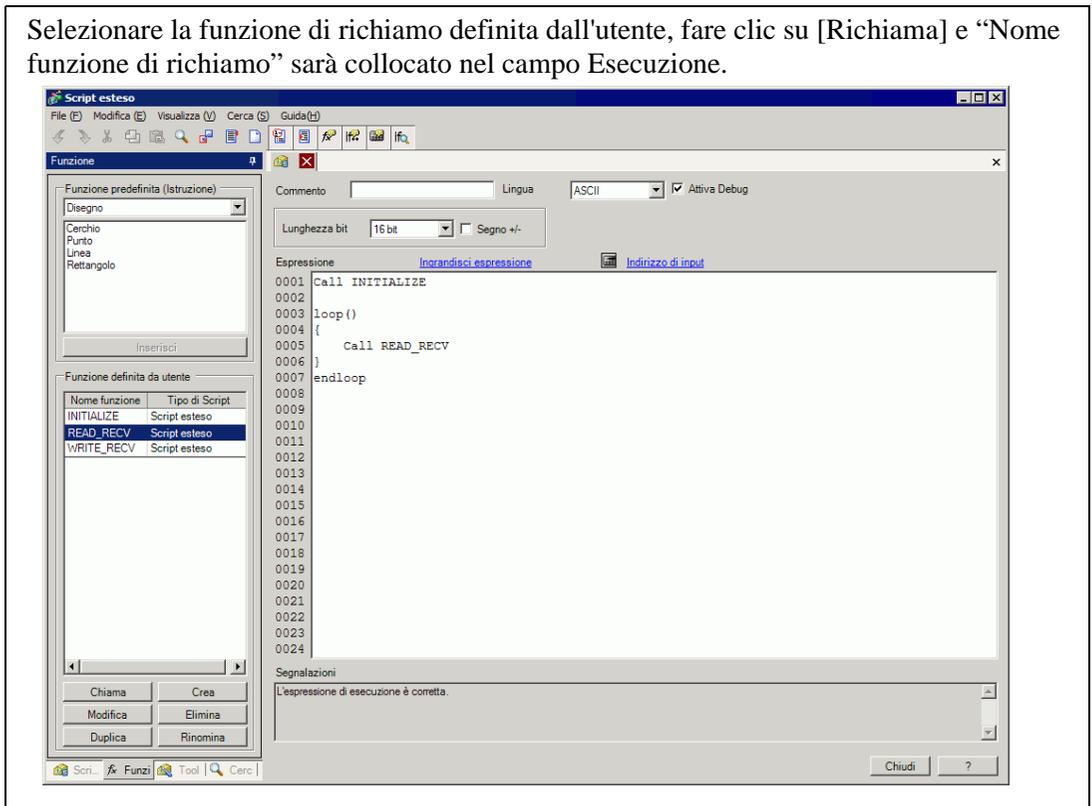
Inserire il nome della funzione e creare lo script nel campo Esecuzione. Fare clic su [OK] e la funzione definita dall'utente sarà salvata.



#### NOTA

- Sono applicabili alcune limitazioni ai nomi delle funzioni. Per altri dettagli, vedere "21.10.3 Limitazioni per le funzioni definite dall'utente" (pagina 21-65)

Selezionare la funzione di richiamo definita dall'utente, fare clic su [Richiama] e “Nome funzione di richiamo” sarà collocato nel campo Esecuzione.



## 21.8 Configurazione condizione trigger

Uno script creato può utilizzare qualsiasi dei seguenti 7 tipi di condizioni trigger.

Impostazione		Descrizione
Azione continua		Lo script viene attivato regolarmente.
Timer		Lo script viene attivato dopo il trascorrere di un periodo designato.
Bit	Bit ON	Quando l'unità GP rileva che il bit designato passa da 0 a 1, lo script viene attivato.
	Bit OFF	Quando l'unità GP rileva che il bit designato passa da 0 a 1, lo script viene attivato.
	Variazione bit	Quando l'unità GP rileva che il bit designato passa da 0 a 1 o da 1 a 0, lo script viene attivato.
Espressioni condizionali	Quando la Condizione è vera	Quando l'unità GP rileva "vero" per un'espressione designata, lo script sarà eseguito.
	Quando la Condizione è "falsa"	Quando l'unità GP rileva "falso" per un'espressione designata, lo script sarà eseguito.

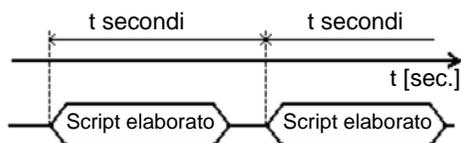
### 21.8.1 Azione continua

Esegue ogni tempo di scansione del visualizzatore.

### 21.8.2 Timer

#### ■ Timer

Ogni volta che il tempo designato scade, lo script viene eseguito una sola volta. La durata del timer può essere impostata da 1 a 32767 secondi.



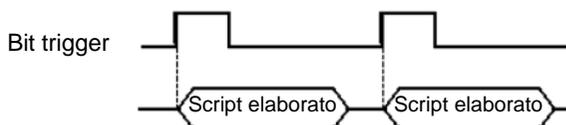
#### NOTA

- Quando si imposta il tempo della funzione timer, il valore temporale includerà il tempo impostato + l'errore temporale di scansione della visualizzazione. Inoltre, sulla base del tempo impiegato per disegnare un elemento dello schermo o per stampare dati, la funzione timer potrà essere rallentata. Per altre informazioni sulla funzione Visualizza tempo di scansione, fare riferimento a " ■ Limitazioni sul bit attivato mediante trigger" (pagina 21-50).
- Quando si usano i D-Script, passare da uno schermo all'altro causerà il riavvio della funzione del timer, partendo da 0.

### 21.8.3 Bit

#### ■ Bit ON

Quando l'unità GP rileva che l'indirizzo bit designato (bit trigger) passa da 0 a 1, lo script viene attivato.

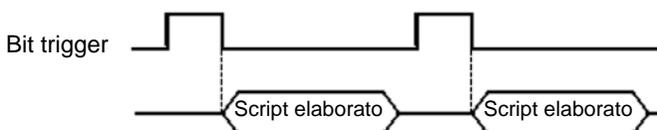


#### NOTA

- Per ON/OFF del bit di trigger, assicurarsi di lasciare il più lungo tra un intervallo più lungo del tempo del ciclo di comunicazione o del tempo di scansione dello schermo. Per altre informazioni su questa funzione, fare riferimento a " ■ Limitazioni sul bit attivato mediante trigger" (pagina 21-50).

#### ■ Bit OFF

Quando l'unità GP rileva che l'indirizzo bit designato (bit trigger) passa da 1 a 0, lo script viene attivato.

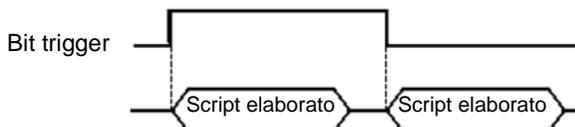


#### NOTA

- Per ON/OFF del bit di trigger, assicurarsi di lasciare il più lungo tra un intervallo più lungo del tempo del ciclo di comunicazione o del tempo di scansione dello schermo. Per altre informazioni su questa funzione, fare riferimento a " ■ Limitazioni sul bit attivato mediante trigger" (pagina 21-50).

#### ■ Cambio di bit

Quando l'unità GP rileva che l'indirizzo bit designato (bit trigger) passa da 0 a 1 o da 1 a 0, lo script viene attivato.



#### NOTA

- Per ON/OFF del bit di trigger, assicurarsi di lasciare il più lungo tra un intervallo più lungo del tempo del ciclo di comunicazione o del tempo di scansione dello schermo. Per altre informazioni su questa funzione, fare riferimento a " ■ Limitazioni sul bit attivato mediante trigger" (pagina 21-50).

## 21.8.4 Espressioni condizionali

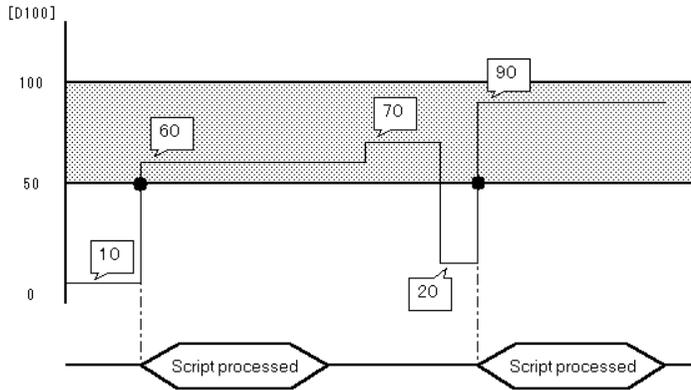
### ■ Quando la Condizione è "vera"

Quando l'unità GP valuta la condizione di attivazione mediante trigger come "vera", lo script viene eseguito una sola volta.

Quando la Condizione trigger è impostata su  $100 > [D100] > 50$ , o script sarà eseguito con la seguente sequenza temporale.

[False] → [True] viene rilevato, lo script viene eseguito, e 70 viene assegnato a D100.

Lo script non viene eseguito quando [True] → [True].



#### NOTA

- Per la condizione di trigger, assicurarsi di lasciare come intervallo il più lungo tra il periodo del ciclo di comunicazione e il tempo di scansione dello schermo. Per altre informazioni su questa funzione, fare riferimento a " ■ Limitazioni sul bit attivato mediante trigger" (pagina 21-50).

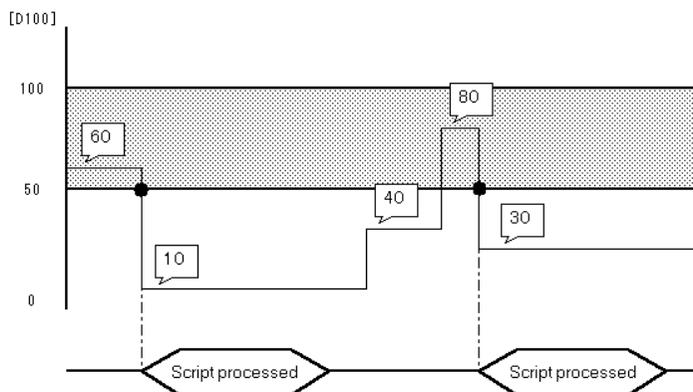
■ **Quando la Condizione è "falsa"**

Quando l'unità GP rileva "falso" per un'espressione designata in un programma di triggering, lo script sarà eseguito una volta sola.

Quando la Condizione trigger è impostata su  $100 > [D100] > 50$ , o script sarà eseguito con la seguente sequenza temporale.

[False] → [True] viene rilevato, lo script viene eseguito, e 20 viene assegnato a D100.

Lo script non viene eseguito quando [True] → [True].



**NOTA**

- Per la condizione di trigger, assicurarsi di lasciare come intervallo il più lungo tra il periodo del ciclo di comunicazione e il tempo di scansione dello schermo. Per altre informazioni su questa funzione, fare riferimento a " ■ Limitazioni sul bit attivato mediante trigger" (pagina 21-50).

## ■ Limitazioni sul bit attivato mediante trigger

- Assicurarsi di lasciare un intervallo più lungo del tempo del ciclo di comunicazione per eseguire operazioni di scrittura sul dispositivo collegato. Quando le operazioni di scrittura sul dispositivo collegato vengono eseguite di frequente utilizzando il timer di scansione del relé speciale interno dell'unità GP, si possono verificare errori di comunicazione o di sistema.
- Quando il bit utilizzato per la Condizione trigger del D-Script viene impostato per il "tocco" e il bit si porta su OFF durante l'elaborazione del D-Script, la sequenza temporale utilizzata premendo ripetutamente l'area di tocco può impedire il rilevamento dell'ascesa del bit. Il trigger del D-Script confronta il valore precedentemente estrapolato a quello attuale per determinare se il trigger ora è "Vero". In ogni caso, durante una scansione singola, il valore memorizzato nell'indirizzo bit ed usato durante l'operazione attivata dal trigger viene mantenuto uguale, anche se viene cambiato durante l'esecuzione. Il nuovo valore viene letto solo dopo l'inizio della scansione successiva.

Tempo del ciclo di comunicazione: L'ora del ciclo di comunicazione è l'ora da quando l'unità di visualizzazione chiede dati al dispositivo/PLC fino a quando li riceve. Il tempo viene memorizzato sotto forma di dati binari nel LS203 del dispositivo interno. L'unità è il millisecondo (ms). La varianza è di +/-10 ms.

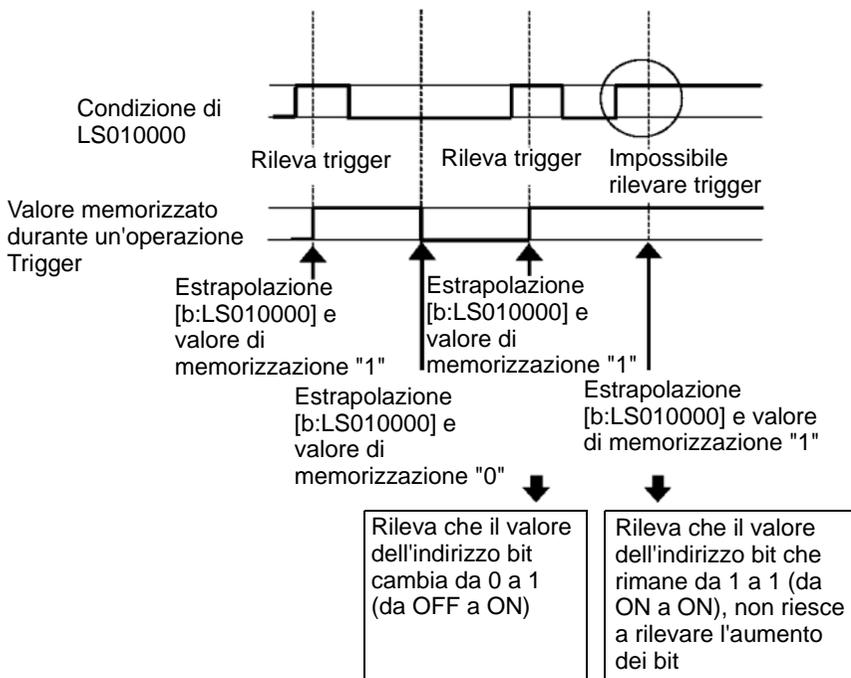
Visualizza il tempo di scansione: Il Tempo di visualizzazione scansione è il periodo richiesto per elaborare uno schermo. Viene memorizzato sotto forma di dati binari nel LS2036 del dispositivo interno. L'unità è il millisecondo (ms). La varianza è di +/-10 ms.

Ad esempio, quando il Tocco viene usato per portare il bit di trigger su ON (LS010000), e il D-Script porta il valore su OFF:

Condizione attivata [#INTERNAL]LS010000

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

◆ **Grafico temporale di elaborazione D-Script**



Ad esempio, se la sequenza temporale del tocco del D-Script non viene utilizzata, e viene eseguito solo il rilevamento, l'elaborazione sarà la seguente.

Uso di un'istruzione if ( ) per rilevare un trigger

Usare un'istruzione per determinare se un'operazione di tocco imposta il bit. Ogni volta che viene eseguita, l'istruzione if ( ) legge il valore ed esegue un'operazione di confronto.

Condizione attivata Bit ON ([#INTERNAL]LS203800 \*1)

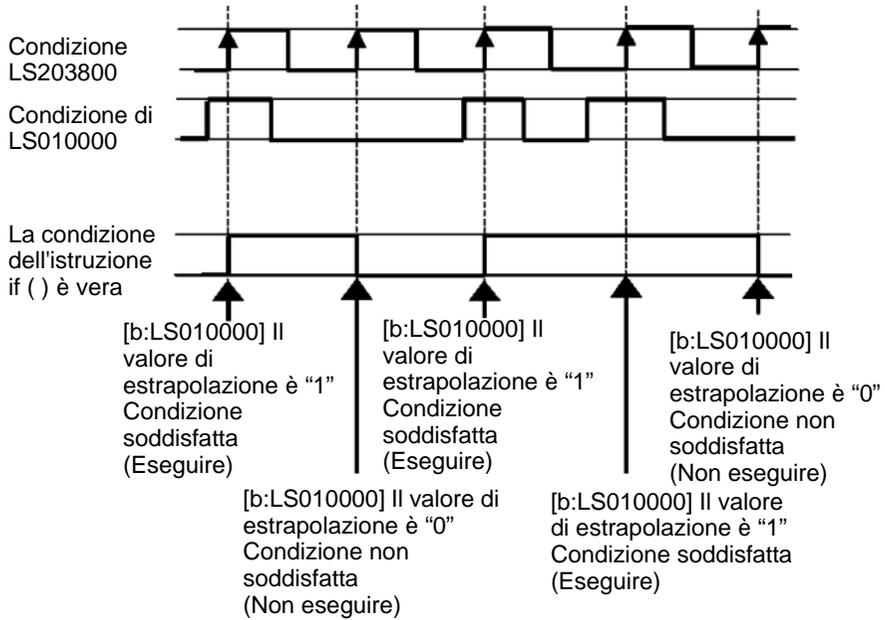
Espressione esecuzione: if ([b:[#INTERNAL]LS010000]==1)

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

\*1 Timer interno dell'unità GP. Il timer incrementa i propri numeri ogni volta che si elabora la Parte impostata sullo schermo di visualizzazione.

Quando si usa il D-script precedente, anche se si inseriscono tocchi consecutivi, lo script verrà eseguito solo se la condizione corrisponde. Come mostrato nel seguente grafico di intervallo temporale, ciascun display esegue una scansione del valore, che viene letto e verificato per trovare una corrispondenza; se la corrispondenza esiste, indipendentemente dal valore precedente, lo script sarà eseguito.

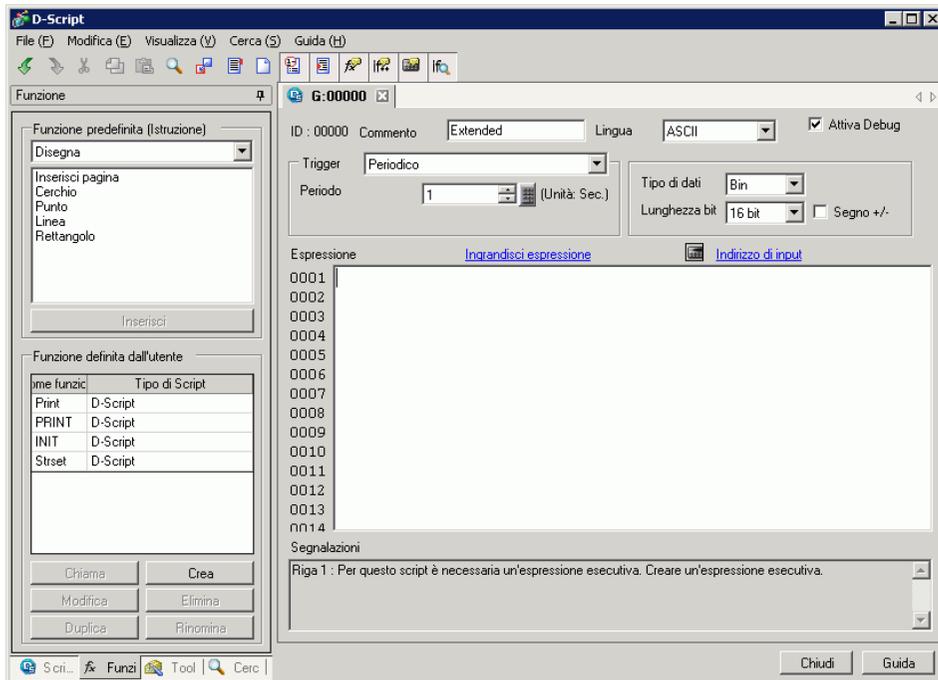
◆ **Grafico temporale di elaborazione D-Script**



## 21.9 Guida alle impostazioni

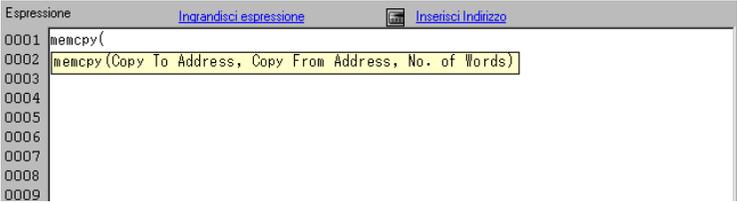
### 21.9.1 Guida alle impostazioni D-Script/[D-Script globale] comune

La seguente finestra di dialogo è quella del [D-Script globale] comune. Le impostazioni da specificare per il D-Script sono uguali a quelle nella finestra di dialogo. L'ID e le impostazioni di trigger non sono specificate per uno [Script esteso] comune: le altre impostazioni, però, sono le stesse.



Impostazione	Descrizione
Esporta	Questo può essere selezionato dal menu Archivio (File). Esporta scrive uno script creato in un file di testo (.txt) che può essere importato in altri script.
Importa	Questo può essere selezionato dal menu Archivio (File). Importa legge in uno script esportato (file di testo).
Numero di riga 	Mostra il numero di righe a destra del programma.
Controllo rientro automatico 	<p>Aggiunge un rientro automatico a istruzioni come quelle indicate di seguito.</p> <pre> Espressione 0001 if ([b:[PLC1]D00000.0]==1) 0002 { 0003     if ( [b:[PLC1]D00001.0]) 0004     [ 0005         [b:[PLC1]D00002.0] 0006     ] 0007     endif 0008 } 0009 endif                     </pre>

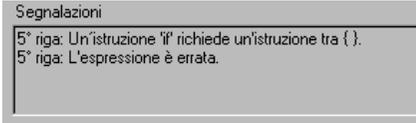
Continua

Impostazione	Descrizione
<p>Assistenza input funzione </p>	<p>Quando la funzione e la parentesi iniziale "(" sono inserite come di seguito, sarà visualizzato il formato della funzione.</p> 
<p>Completamento automatico sintassi </p>	<p>Quando "if" o "loop" sono inseriti con il tastierino, la sintassi rimanente viene sistemata automaticamente.</p>

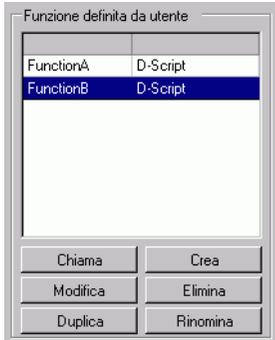
Continua

Impostazione	Descrizione								
<p data-bbox="112 909 252 971">Indirizzo di input</p> 	<p data-bbox="384 171 1218 233">Quando si crea uno script, inserire una parentesi quadra sinistra ( [ ) per visualizzare la finestra di dialogo [Input indirizzo].</p> <div data-bbox="584 253 1055 417" style="text-align: center;"> </div> <p data-bbox="384 436 1163 498">Selezionare il tipo di indirizzo tra [Indirizzo bit], [Indirizzo word], [Indirizzo temporaneo].</p> <ul data-bbox="384 508 1207 774" style="list-style-type: none"> <li>• <b>Indirizzo bit</b> Si può specificare l'indirizzo del dispositivo/PLC, l'indirizzo interno dell'unità GP e la variabile bit.</li> <li>• <b>Indirizzo word</b> Si può specificare l'indirizzo del dispositivo/PLC, l'indirizzo interno dell'unità GP e la variabile intera.</li> <li>• <b>Indirizzo provvisorio</b> Quest'indirizzo può essere utilizzato solo per gli script.</li> </ul> <p data-bbox="384 797 976 826">Fare riferimento ai dettagli del dispositivo interno.</p> <p data-bbox="384 832 1234 890">☞ "A.1.2 Comunicazione con un dispositivo/PLC utilizzando il metodo Accesso diretto" (pagina A-4)</p> <p data-bbox="384 896 1230 954">☞ "A.1.3 Utilizzo del metodo Link memoria con Dispositivi/PLC non supportati" (pagina A-7)</p> <p data-bbox="384 977 518 1006"><b>IMPORTANTE</b></p> <ul data-bbox="384 1020 1223 1141" style="list-style-type: none"> <li>• Negli script NON impostare alcuna password, ecc., che inizi per "0". Tutti i valori numerici che iniziano per "0" saranno elaborati come dati Ott (base-8).</li> <li>• Come descrivere diversi formati in inserimento dati</li> </ul> <p data-bbox="384 1151 536 1180">Ad esempio:</p> <ul data-bbox="445 1193 1093 1418" style="list-style-type: none"> <li>• DEC (Base-10) : Valore iniziale non uguale a zero Ad esempio, 100</li> <li>• Esad (Base-16) : Valore che inizia per 0x Ad esempio, 0x100</li> <li>• OTT (Base-8) : Valore che inizia per 0x Ad esempio, 0100</li> </ul> <ul data-bbox="384 1447 1245 1508" style="list-style-type: none"> <li>• Esempio di operazione con diversi formati di dati utilizzando l'operatore AND (Esad e BCD)</li> </ul> <table data-bbox="445 1528 989 1673" style="margin-left: 40px;"> <tr> <td>Esad solo</td> <td></td> </tr> <tr> <td>0x270F &amp; 0xFF00</td> <td>Risultato: 0x2700</td> </tr> <tr> <td>BCD ed Esad.</td> <td></td> </tr> <tr> <td>9999 9999 &amp; 0xFF00</td> <td>Risultato: 0x9900</td> </tr> </table>	Esad solo		0x270F & 0xFF00	Risultato: 0x2700	BCD ed Esad.		9999 9999 & 0xFF00	Risultato: 0x9900
Esad solo									
0x270F & 0xFF00	Risultato: 0x2700								
BCD ed Esad.									
9999 9999 & 0xFF00	Risultato: 0x9900								

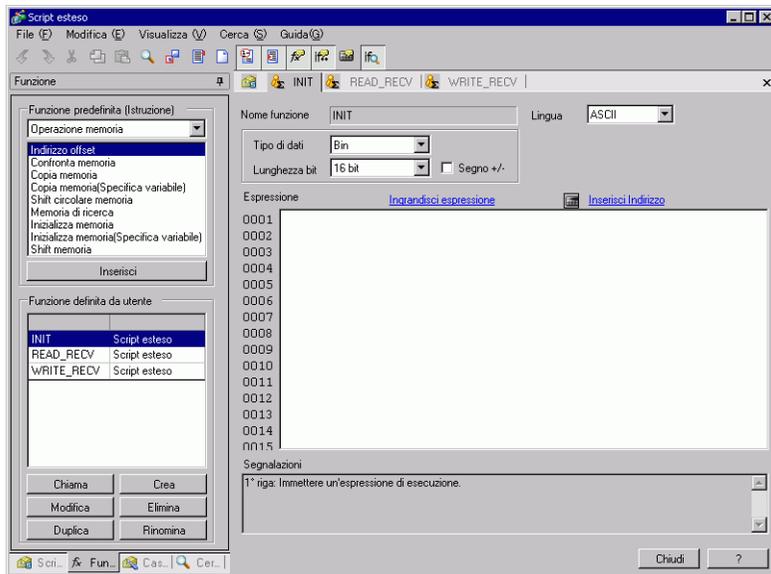
Continua

Impostazione	Descrizione
<p>Analisi automatica sintassi </p>	<p>Controlla la sintassi durante la creazione dello script. I risultati dei controlli saranno visualizzati nella parte inferiore della finestra.</p> 
<p>ID</p>	<p>Gli script sono gestiti da un numero ID. Quando si creano più script con diverse condizioni di trigger, impostare un valore da 0 a 65535.</p>
<p>Commento</p>	<p>Inserire un commento per lo script.</p>
<p>Lingua</p>	<p>Scegliere una lingua dall'elenco a discesa: [ASCII], [Giapponese], [Cinese (Tradizionale)], [Cinese (Semplificato)] o [Coreano].</p>
<p>Attiva funzione debugging</p>	<p>Impostare se attivare o no la funzione di debugging. Se la funzione _debug esiste nel corpo dello script, sarà eseguita. Per altre informazioni su questa funzione, fare riferimento a " ■ Funzione debugging" (pagina 21-142).</p>
<p>Blocca password</p>	<p>Specifica se attivare la funzione Blocca password.</p> <ol style="list-style-type: none"> <li>(1) Selezionare la casella per visualizzare lo schermo [Impostazioni livello utente].</li> <li>(2) Selezionare il livello utente tra [Utente standard], [Power User] e [Utente con privilegi avanzati], quindi fare clic su [OK].</li> <li>(3) Inserire la [Password] e fare clic su [OK]. Quando si seleziona la casella di controllo [Memorizza password], la password viene salvata e la finestra di dialogo Sblocca password non viene visualizzata.</li> </ol>  
<p>Trigger</p>	<p>Impostare la condizione di trigger che causerà l'esecuzione dello script. Per altre informazioni su questa funzione, fare riferimento a "21.8 Configurazione condizione trigger" (pagina 21-46). Gli script estesi non hanno l'impostazione per la condizione trigger.</p>
<p>Tipo di dati</p>	<p>Impostare il formato dei dati per lo script su Bin o BCD. Per gli script estesi, Bin è fisso.</p>
<p>Lunghezza bit</p>	<p>Impostare la lunghezza dei dati per lo script a 16 o 32 bit.</p>
<p>Segno +/-</p>	<p>Effettuare questa selezione quando si desiderano inserire i numeri negativi. Quest'opzione può essere impostata solo quando il Tipo di dati è Bin.</p>
<p>Espressione esecuzione</p>	<p>Contenuti dello script.</p>

Continua

Impostazione	Descrizione
<p>Funzione incorporata (Istruzione)</p>	<p>Nella barra strumenti, selezionare comandi e funzioni per aggiungerli più facilmente allo script.</p> <p>Per altre informazioni su comandi e funzioni disponibili da usare, vedere  "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)</p> <p>Funzioni incorporate</p>  <p>Selezionare una categoria da [Funzione incorporata (Istruzione)]. Le relative funzioni appariranno nell'area di fondo.</p> <p>Selezionare la funzione e fare clic su [Input]: apparirà la finestra di dialogo Impostazioni corrispondente.</p>
<p>Funzioni definite dall'utente</p>	<p>Registrare uno script creato come funzione definita dall'utente e che potrà essere usato da altri script.</p> <div data-bbox="386 823 477 865" style="border: 1px solid black; padding: 2px; margin: 10px 0;"><b>NOTA</b></div> <ul style="list-style-type: none"> <li>• Per ulteriori dettagli sulle funzioni definite dall'utente, vedere "21.9.2 Guida alle impostazioni delle funzioni definite dall'utente" (pagina 21-58).</li> </ul> 
<p>Toolbox</p>	<p>Come menu di selezione rapida, selezionare comandi del Toolbox da usare nello script.</p> <p>Inoltre, si possono selezionare comandi utilizzati negli script, come Cerca e Colloca testo usato negli script.</p> <p>Per altre informazioni sui comandi disponibili, vedere "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)</p> 

## 21.9.2 Guida alle impostazioni delle funzioni definite dall'utente



Impostazione	Descrizione
Nome funzione	Visualizza i nomi delle funzioni definite dall'utente.
Tipo di script	Visualizza il tipo di script. Dal menu a discesa è possibile passare dal [D-Script] allo [Script esteso] e viceversa.
Richiamare	Richiamare una funzione creata. Selezionare la funzione da richiamare, fare clic su [Richiama] e “Richiama Nome funzione” sarà collocato nel campo Esecuzione.
Crea	Creare una nuova funzione. Fare clic su [Crea]. Apparirà la finestra di dialogo [Nome funzione].
Modifica (E)	Modificare una funzione esistente. Selezionare una funzione da modificare, quindi fare clic su [Modifica]. Apparirà la finestra di dialogo [Funzione D-Script].
Elimina	Eliminare una funzione esistente. Selezionare la funzione da eliminare e fare clic su [Elimina].
Duplica	Copiare una funzione esistente. Selezionare la funzione da copiare e fare clic su [Copia] per visualizzare la finestra di dialogo da usare per creare il nome della copia della funzione.
Rinomina	Cambiare il nome di una funzione esistente. Fare clic su [Rinomina]. Appare la finestra di dialogo Rinomina funzione.

## 21.10 Limitazioni

### 21.10.1 Limitazioni per il D-Script/D-Script globale

- Nella programmazione del D-Script, tre indirizzi occupano lo stesso spazio in memoria di una Parte. Il numero massimo di indirizzi disponibili per un D-Script è 255\*<sup>1</sup>. Usare il numero minimo di indirizzi possibile, poiché più dispositivi si usano, più lenta sarà la risposta.
- Il D-Script non è in grado di eseguire calcoli su valori dimensionabili (Variabili dimensionabili o reali). Anche sulle variabili di scrittura non si possono eseguire calcoli. In ogni caso, si possono eseguire calcoli su elementi individuali dalle variabili di struttura.
- Le dimensioni di un D-Script influiscono sul Tempo di scansione del visualizzatore. Nota: l'uso di un gran numero di indirizzi potrebbe deteriorare in modo significativo la prestazione del programma.
- Non specificare [Azione continua] nelle Condizioni trigger per mettere lo script in grado di scrivere negli indirizzi del dispositivo/PLC. Apparirà un errore, poiché la procedura di comunicazione non riesce a tenere il passo con la grande quantità di istruzioni di scrittura. Per attivare l'[Azione continua], usare il dispositivo interno GP o un indirizzo temporaneo.
- Quando si richiama una funzione da una funzione, il massimo numero di livelli di annidamento è 9. Non impostare più di 9 livelli.
- Si possono creare fino a 9 livelli di chiamate annidate.
- Si possono creare fino a 254 funzioni.

\*1 Numero totale di dispositivi usati nelle espressioni trigger e nei programmi di script.

◆ Secondo i dispositivi specificati per le condizioni di trigger, le operazioni di D-Script attivate da un trigger dopo i cambi di schermo sono le seguenti:

Condizione Trigger	Valore o condizione attuale	Qualsiasi dispositivo attivato diverso da [#MEMLINK]				[#MEMLINK]			
		Bit "0"	Bit "1"	Condizione non soddisfatta	Condizione soddisfatta	Bit "0"	Bit "1"	Condizione non soddisfatta	Condizione soddisfatta
Margine principale del bit		X	O	–	–	X	X	–	–
Margine di caduta del bit		O	X	–	–	X	X	–	–
Variazione bit		O	O	–	–	X	X	–	–
Impostazioni timer		X	X	X	X	X	X	X	X
Rilevare "vero"		–	–	X	O	–	–	X	O
Rilevare "falso"		–	–	O	X	–	–	O	X

O: l'operazione viene eseguita immediatamente dopo il cambio di schermo, oppure l'alimentazione viene attivata.

X: l'operazione non viene eseguita immediatamente dopo il cambio di schermo, oppure l'alimentazione viene attivata.

- Quando il timer sta funzionando, inizierà a contare immediatamente dopo il cambio di schermo.
- Quando si usano i D-Script globali, le operazioni menzionate sono eseguite solo quando l'alimentazione dell'unità GP si trova su ON. Quando lo schermo dell'unità GP cambia, comunque, l'operazione menzionata non sarà eseguita e le condizioni di trigger sono monitorate di continuo.
- Quando un D-Script globale include un timer, il timer inizierà a contare non appena l'unità GP viene alimentata.

**NOTA**

- Non usare il tasto del pannello al tocco per impostare il bit di trigger o per far funzionare il bit di avvio in un programma. Il tempo dell'inserimento al tocco potrebbe non essere corretto, e avere come risultato l'inserimento non corretto del bit.

◆ **Quando un valore viene assegnato ad un indirizzo per cambiare schermo mentre un comando D-Script viene eseguito, l'operazione di cambio di schermo viene elaborata dopo tutti i D-Script.**

Ad esempio:

ID	00000			
Tipo di dati	Bin	Lunghezza dati 16 bit	Segno +/-	Nessuno
Trigger	Bit di comando([b:M0000])			
[w:[PLC1]D0100]=0		// (1)		
[w:[#INTERNAL]LS0008]=30		// (2) Passa allo schermo di base n. 30		
[w:[PLC1]D0101]=1		// (3)		
[w:[PLC1]D0102]=2		// (4)		

Quando il D-Script sopra citato viene eseguito, l'elaborazione del cambio di schermi viene eseguita dopo quella di (3) e (4).

◆ **Quando i dati utilizzati in un D-Script sono configurati con un'operazione al tocco dell'unità GP, verificare che l'operazione di scrittura dati sia completata prima di eseguire il D-script.**

◆ **Limitazioni specifiche per il D-Script globale**

- Quando l'alimentazione dell'unità GP si trova su ON, saranno eseguite le azioni mostrate nella tabella alla pagina precedente. Al momento di cambiare lo schermo, la tabella sopra non sarà applicata, e le condizioni di trigger saranno continuamente controllate.
- Il funzionamento del D-Script globale viene sospeso durante i cambi di schermo o altre operazioni di GP.
- Dopo aver avviato l'alimentazione dell'unità GP, le azioni del D-Script globale non saranno eseguite fino al completamento di tutte le letture dei dati per lo schermo iniziale. In ogni caso, dopo i cambiamenti iniziali sullo schermo, le azioni del D-Script globale possono essere eseguite prima che la lettura dei dati sia completa.
- Il numero massimo di dispositivi nei D-Script globali è  $255^{*1}$ . Quando si supera questo numero, il D-Script non funzionerà. Poiché questi dispositivi leggono sempre i dati, indipendentemente dagli schermi, assicurarsi di utilizzare solo il numero minimo di impostazioni del dispositivo nel D-Script. In caso contrario, le prestazioni dell'operazione potrebbero essere degradate.
- Il numero massimo di D-Script globali disponibili è 32; anche la funzione attualmente utilizzata conta come un D-Script globale. Quando il numero di D-Script globali raggiunge 32, qualsiasi D-Script globale seguente sarà ignorato.

\*1 Numero totale di dispositivi usati nelle espressioni trigger e nei programmi di script.

### ◆ Limitazioni per le Operazioni porta SIO

- Gli indirizzi designati nelle funzioni Invia/Ricevi non saranno aggiunti al conteggio degli indirizzi dei D-Script.
- Il Controllo è una variabile di sola scrittura, mentre Stato e Dati ricevuti sono variabili di sola lettura. La lettura della variabile di controllo o dei dati di scrittura nella variabile di stato provoca il fallimento dell'operazione.
- Creare D-Script (o funzioni) indipendenti per inviare e ricevere operazioni. Per ulteriori informazioni sui diagrammi dei trasferimenti dati, fare riferimento a  " ■ Diagramma" (pagina 21-23)
- L'area utente nel dispositivo LS (da LS20 a LS2031 e da LS2096 a LS8191) può inviare dati per le funzioni Invia/Ricevi.
- Nell'area di lavoro [Impostazioni di sistema], pagina [Impostazioni I/O Script], quando il [Tipo] non è impostato su [D-Script/ D-Script globale], il 13mo bit nell'indirizzo LS2032 si porta su ON quando [D-Script/D-Script globale] esegue le funzioni di Impostazione etichetta dell'[Operazione porta SIO] (Invia, Ricevi, Controlla, Leggi stato e Ricevi dimensioni dati). Per informazioni sui relè speciali:  "A.1.4.3 Relè speciale" (pagina A-27)
- Quando si usano le funzioni Invia/Ricevi, impostare la lunghezza del D-Script su 16 bit. Notare che l'operazione non riuscirà se la lunghezza del bit è impostata su 32.
- Le dimensioni del buffer Invia consistono in 2048 byte, mentre quelle del bit Ricevi consistono in 8192 byte. Il segnale ER (emissione) segnale RS (emissione) viene portato su OFF dopo che almeno l'80% del buffer Ricevi è stato riempito con dati ricevuti.

### ◆ Limitazioni per le operazioni in formato BCD

Se un valore che non può essere convertito in formato BCD viene trovato durante il funzionamento, il programma arresterà l'esecuzione.

Questi valori sono compresi da A a F in formato esadecimale.

Non utilizzare questi valori. Se il programma si interrompe a causa di valori non-BCD, il bit 7 nell'informazione del relé comune (LS2032) nell'unità GP si porterà su ON. Questo bit non si porterà su OFF fino a quando l'unità GP si spegne o si porta in modalità Offline.

Ad esempio:

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

Se D300 è 3, spostandosi a sinistra di due bit si otterrà 0x000C, che non può essere convertito in un formato BCD, ed interrompe l'esecuzione del programma.

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

Se D300 è 3, spostandosi di due bit a sinistra si ottiene 0x000C.

A differenza dell'esempio illustrato sopra, 0x000C è il risultato dell'operazione da salvare in memoria, e non causa l'arresto del programma.

### ◆ Limitazioni per le Operazioni con zeri

Se si divide per zero nelle operazioni di divisione (/) e di modulo (%) l'esecuzione si arresterà. Non dividere per zero.

Se il programma si interrompe a causa dell'errore precedente, il bit 8 del relé comune (LS2032) nell'unità GP si porterà su ON. Questo bit non si porterà su OFF fino a quando l'unità GP si spegne o si porta in modalità Offline.

**◆ Note sul ritardo durante Assegna operazione**

L'uso di un indirizzo di dispositivo in un'operazione di assegnazione può causare ritardi nella scrittura, poiché l'unità GP deve leggere i dati dell'indirizzo provenienti dal dispositivo collegato. Considerare quanto segue.

Ad esempio:

$$[w:[PLC1]D0200]=[w:[PLC1]D0300]+1 \dots$$
$$[w:[PLC1]D0201]=[w:[PLC1]D0200]+1 \dots$$

L'istruzione (1) assegna (D0300+1) in D0200. In ogni caso, nell'istruzione (2), il risultato dell'istruzione (1) non è stato assegnato in D0200 a causa di lunghe comunicazioni con il dispositivo/PLC. In questi casi, programmare in modo che il risultato dell'istruzione (1) sia memorizzato nell'area LS prima dell'esecuzione, come illustrato di seguito.

$$[w:[\#INTERNAL]LS0100]=[w:[PLC1]D0300]+1$$
$$[w:[PLC1]D0200] = [w:[\#INTERNAL]LS0100]$$
$$[w:[PLC1]D0201]=[w:[\#INTERNAL]LS0100]+1$$
**◆ Note sulle operazioni con numeri negativi**

Per le funzioni in cui un numero negativo viene inserito per un argomento che non accetta questo tipo di numeri <sup>\*1</sup>, il numero inserito funzionerà come <sup>\*2</sup> senza segni.

\*1 Ad esempio, il "numero di byte" dell'argomento `_CF_read ()` non può accettare numeri negativi perché è la dimensione dei dati ad essere letta.

\*2 Ad esempio, -1 è gestito come 65535 per 16 bit, e 4294967295 per 32 bit.

## 21.10.2 Limitazioni per gli script estesi

- Per gli indirizzi di dispositivo, potranno essere utilizzate solo l'area LS e l'area USR (Area utente estesa).
- Gli indirizzi temporanei dei D-Script e dei D-Script globali sono gestiti in modo indipendente dall'indirizzo temporaneo degli Script estesi. Pertanto, le modifiche apportate agli indirizzi temporanei dei D-Script e dei D-Script globali non saranno rispecchiate nell'indirizzo temporaneo degli Script estesi.
- Si possono richiamare le funzioni create con D-Script/ D-Script globale, ma se si accede a un indirizzo di dispositivo esterno al dispositivo interno nella funzione, potrebbe non funzionare normalmente. Inoltre, quando vengono trasferite (durante la creazione di dati per l'unità GP), le funzioni definite dall'utente saranno create indipendentemente per i D-Script, i D-Script globali e gli Script estesi.
- Quando si richiama una funzione da una funzione, il massimo numero di livelli di annidamento è 9.
- Si possono richiamare fino a 254 funzioni. (Il numero di funzioni disponibile con "Richiamo" è 254).
- Lo script esteso non influisce sul conteggio dei tag.
- Le funzioni supportate solo dallo script esteso, ad esempio, le operazioni stringa, non funzioneranno se richiamate con i D-Script o i D-Script globali.
- Il formato dati disponibile è Bin. Il formato dati BCD è disattivato.
- Le dimensioni del buffer Invia consistono in 2048 byte, mentre quelle del bit Ricevi consistono in 8192 byte. La linea CTS viene spenta dopo che circa l' 80% del buffer Ricevi viene riempito con i dati ricevuti.
- Non si possono selezionare simultaneamente D-Script/D-Script globali e Script estesi. Notare le combinazioni elencate nella tabella riportata di seguito.

Impostazione SIO esteso	D-Script/D-Script globale Funzione SIO estesa per uno script esteso	Funzione SIO estesa per uno script esteso
D-Script/D-Script globale	O: Operazione possibile	X: Non funzionerà
Script esteso	X: Non funzionerà	O: Operazione possibile

- Convenzioni notazionali per l'impostazione delle stringhe dei caratteri

Quando si usano stringhe caratteri con "\_ strset ()" ed altre funzioni, includere la stringa carattere in virgolette doppie ("). Per visualizzare le virgolette doppie nelle stringhe caratteri, apporre il simbolo "\" ed esprimerlo come [\"]. Non esiste un modo per rappresentare un singolo simbolo "\". Se necessario, usare l'impostazione del formato codice carattere (\_strset (databuf0, 92)).

Ad esempio:

```
"ABC\"DEF"   ABC"DEF
"ABC\DEF"    ABC\DEF
"ABC\\"DEF"  ABC\\"DEF
"ABC\\DEF"   ABC\\DEF
```

- Per le funzioni in cui un numero negativo viene inserito per un argomento che non accetta questo tipo di numeri \*<sup>1</sup>, il numero inserito funzionerà come \*<sup>2</sup> senza segni.

\*1 Ad esempio, il "numero di byte" dell'argomento `_CF_read ()` non può accettare numeri negativi perché è la dimensione dei dati ad essere letta.

\*2 Ad esempio, `-1` è gestito come `65535` per 16 bit, e `4294967295` per 32 bit.

◆ **Dimensioni dei buffer dedicati per SIO esteso, databuf0, databuf1, databuf2 e databuf3.**

Buffer	Nome buffer	Dimensioni
Buffer dati 0	databuf0	1 KB
Buffer dati 1	databuf1	1 KB
Buffer dati 2	databuf2	1 KB
Buffer dati 3	databuf3	1 KB

**21.10.3 Limitazioni per le funzioni definite dall'utente**

- Alcune parti dei comandi da utilizzare differiscono ad ogni script. Quando si usano i comandi, fare riferimento a "21.11 Comandi programma/Espressioni condizionali" (pagina 21-69)
- Per il nome funzione, si può usare qualsiasi lettera dell'alfabeto inglese o il carattere di sottolineatura "\_" (Comunque, il nome della funzione deve iniziare con un carattere alfanumerico.)
- Non usare quanto segue come Nomi Funzione.

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

## 21.10.4 Note sui risultati dell'operazione

---

### ■ Cifre in eccesso

Qualsiasi cifra in eccesso risultante dalle operazioni sarà troncata.

Quando si esegue un'operazione sui dati a 16 bit senza segni:

- $65535 + 1 = 0$  (Produce cifre in eccesso)
- $(65534 * 2) / 2 = 32766$  (Produce cifre in eccesso)
- $(65534 / 2) * 2 = 65534$  (Non produce cifre in eccesso)

### ■ Differenza dell'elaborazione residua

Il risultato dell'elaborazione residua dipende dal segno o dalla mancanza di un segno sui lati destro e sinistro.

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

### ■ Cifre decimali troncate

I valori frazionali risultanti dalla divisione sono troncati.

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

### ■ Note sui dati BCD funzionanti

Un'operazione dati BCD che produce cifre in eccesso non dà il risultato corretto.

## 21.10.5 Errori

Il seguente messaggio di errore viene visualizzato quando uno script è configurato in modo non corretto. L'errore sarà visualizzato in fondo allo schermo dell'unità GP.

I codici di errore sono scritti negli indirizzi LS91XX. Il numero scritto nell'area del codice di errore sarà la parte numerica che segue RAAA nella tabella riportata di seguito. (Ad esempio, quando si verifica l'errore RAAA13, sarà scritto '130'.)

Elenco dei codici d'errore

D-Script (Indirizzo errore=LS9120)	D-script globale (Indirizzo errore=LS9110)	Script esteso (Indirizzo errore=LS9100)
–	RAAA130	RAAA140
Inutilizzato	Errore D-Script globale. (Il numero totale di D-Script globali supera il limite massimo di 32).	Errore D-Script esteso (Il numero totale delle funzioni supera il numero massimo consentito di 255.)
–	RAAA131	–
Inutilizzato	Errore D-Script globale. (Il n. totale di dispositivi supera il limite massimo di 255 <sup>*1</sup> ..)	Inutilizzato
RAAA120	RAAA132	RAAA141
Errore D-Script (La funzione specificata non esiste o la funzione contiene un errore.)	Errore D-Script esteso (La funzione specificata non esiste oppure contiene un errore.)	Errore D-Script esteso (La funzione specificata non esiste oppure contiene un errore.)
RAAA121	RAAA133	RAAA142
Errore D-Script (Queste funzioni sono annidate in 10 o più livelli.)	Errore D-Script esteso (Queste funzioni sono annidate fino a 10 livelli od oltre.)	Errore D-Script esteso (Queste funzioni sono annidate fino a 10 livelli od oltre.)
RAAA122	RAAA134	RAAA143
Errore del D-Script (esiste un'espressione non supportata da questa versione.)	Errore D-Script globale (esiste un'espressione che non è supportata da questa versione.)	Errore D-Script esteso (Esiste un'espressione non supportata da questa versione.)
RAAA123	RAAA135	RAAA144
Errore D-Script (la funzione di operazione SIO viene utilizzata in una condizione in cui non sono stati impostati dispositivi/PLC.)	Errore D-Script esteso (La funzione di operazione SIO viene utilizzata in una condizione dove non è stato impostato alcun dispositivo/PLC.)	Errore D-Script esteso (La funzione di operazione SIO viene utilizzata in una condizione dove non è stato impostato alcun dispositivo/PLC.)

Continua

RAAA124	RAAA136	RAAA145
Il D-Script presenta un errore.	Il D-Script globale presenta un errore.	Il D-Script esteso presenta un errore.

\*1 Numero totale di dispositivi usati nelle espressioni trigger e nei programmi di script.

## 21.11 Comandi programma/Espressioni condizionali

### ■ Funzione

Elemento	Comando/Funzione	D-Script/D-Script globale	Script esteso
Tipo di dati	Bin, BCD	O	solo Bin
Lunghezza bit	16 bit, 32 bit	O	O
Segno +/-	Attivato/Disattivato	O	O
Trigger	Impostazioni timer	O	X
	Margine principale del bit	O	X
	Margine di caduta del bit	O	X
	Alterna bit	O	X
	L'espressione è vera	O	X
	L'espressione è falsa	O	X
Disegna	Carica schermo	O	X
	Dot	O	O
	Linea	O	O
	Cerchio	O	O
	Rettangolo	O	O
Operatore	Aggiunta (+)	O	O
	Sottrazione (-)	O	O
	Modulo (%)	O	O
	Moltiplicazione (*)	O	O
	Divisione (/)	O	O
	Assegnazione (=)	O	O
Confronto	AND logico	O	O
	OR logico	O	O
	Negazione (NOT)	O	O
	Inferiore a (<)	O	O
	Inferiore o uguale a (<=)	O	O
	Non uguale a (<>)	O	O
	Superiore a (>)	O	O
	Superiore o uguale a (>=)	O	O
	Uguale a (==)	O	O

Continua

Elemento	Comando/Funzione	D-Script/D-Script globale	Script esteso
Operazione memoria	Copia memoria: memcpy ( )	O	O
	Inizializza memoria: memset ( )	O	O
	Copia memoria (Specificazione variabili) _memcpy_EX ( )	O	O
	Inizializza memoria (Specificazione variabili) _memset_EX ( )	O	O
	Indirizzo di specifica valore offset	O	O
	Sposta memoria	O	O
	Ring memoria	O	O
	Cerca memoria	O	O
	Confronta memoria	O	O
Operazione su bit	Sposta verso sinistra (<<)	O	O
	Sposta verso destra (>>)	O	O
	Bitwise AND (&)	O	O
	Bitwise OR ( )	O	O
	Bitwise XOR (^)	O	O
	Complemento a 1	O	O
	Bit impostato: set ( )	O	O
	Azzerà bit: clear()	O	O
	Alterna bit: alterna ( )	O	O
Espressioni condizionali	if ( )	O	O
	if ( ) else	O	O
	loop ( ), break	O	O
	loop ( ) loop infinito	X	O
Indirizzo	Indirizzo bit	O	Dispositivo interno
	Indirizzo word	O	Dispositivo interno
	Indirizzo funzionante temporaneo	O	O*1
Costante	Dec, Esad, Ott	O	O

Continua

Elemento	Comando/Funzione	D-Script/D-Script globale	Script esteso
Funzione SIO	Ricevi IO_READ ([p:SIO])	O	O
	Invia: IO_WRITE ([p:SIO ])	O	O
	Ricezione estesa: _IO_READ_EX ( )	X	O
	Invia esteso _IO_WRITE_EX ( )	X	O
	Funzione di ricezione in standby _IO_READ_WAIT ( )	X	O
	Controllo [c:EXT_SIO_CTRL]	O	O
	Stato [s:EXT_SIO_STAT]	O	O
	Dimensioni dati ricevuti [r:EXT_SIO_RCV]	O	O
	Pausa: _wait ( )	X	O

Elemento	Comando/Funzione	D-Script/D-Script globale	Script esteso
Operazione del testo	Testo	X	O
	Buffer dati databuf0, databuf1, databuf2, databuf3	X	O
	Scrivi stringa _strset ( )	X	O
	Dal buffer dati al dispositivo interno _dlcopy ( )	X	O
	Copiare dal dispositivo interno al buffer dati: _ldcopy ( )	X	O
	Conversione esadecimale testo in numero intero _hexasc2bin ( )	X	O
	Conversione testo decimale in intero _decasc2bin ( )	X	O
	Conversione del numero esadecimale in stringa _bin2hexasc ( )	X	O
	Conversione del numero esadecimale in stringa _bin2decasc ( )	X	O
	Lunghezza stringa _strlen ( )	X	O
	Stringa concatenata _strcat ( )	X	O
	Copia stringa parziale _strmid ( )	X	O
	Stato [e:STR_ERR_STAT]	X	O

Continua

Elemento	Comando/Funzione	D-Script/D-Script globale	Script esteso
Funzione	Richiamare	O	O
	return	X	O
Operazione file CF	Leggi file CSV	O	O
	Elenco file output _CF_dir ( )	O	O
	Leggi file _CF_read ( )	O	O
	Leggi file CSV _CF_read ( )	O	O
	Scrivi file _CF_write ( )	O	O
	Elimina file _CF_delete ( )	O	O
	Cambia nome file _CF_rename ( )	O	O
Operazione file USB	File lettura USB	O	O
	Elenco file output _USB_dir ( )	O	O
	Leggi file _USB_read ( )	O	O
	Leggi file CSV USB_read_csv ( )	O	O
	Scrivi file _USB_write ( )	O	O
	Elimina file _USB_delete ( )	O	O
	Cambia nome file _USB_rename ( )	O	O
Operazione stampante	Porta COM di output: IO_WRITE ([p:PRN])	O	O
Debugging	_debug ( )	O	O

\*1 L'indirizzo temporaneo esiste indipendentemente dai D-Script convenzionali e globali.

### 21.11.1 Operazione su bit

Operazione su bit	Riepilogo funzioni
	<b>Impostazioni bit</b>  " ■ Impostazioni bit" (pagina 21-74) Modifica l'indirizzo bit specificato da 0 > a 1.
	<b>Azzera bit</b>  " ■ Azzera bit" (pagina 21-74) Modifica l'indirizzo bit specificato da 1 > a 0.
	<b>Alterna bit</b>  " ■ Alterna bit" (pagina 21-74) Modifica l'indirizzo bit specificato da 1-> a 0 o da 0-> a 1.

#### ■ Impostazioni bit

Elemento	Descrizione
Riepilogo	Modifica l'indirizzo bit specificato da 0 > a 1.
Format	set ( )

#### Espressione esempio:

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

Nell'esempio riportato sopra, il 00mo bit di LS0100 viene portato da 0 -> a 1.

#### ■ Azzera bit

Elemento	Descrizione
Riepilogo	Modifica l'indirizzo bit specificato da 1 > a 0.
Format	clear()

#### Espressione esempio:

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

Nell'esempio riportato sopra, il 00mo bit di LS0100 viene portato da 1 -> a 0.

#### ■ Alterna bit

Elemento	Descrizione
Riepilogo	Modifica l'indirizzo bit specificato da 1 -> a 0 o da 0 -> a 1.
Format	alterna ( )

#### Espressione esempio:

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

Nell'esempio riportato sopra, il 00mo bit di LS0100 viene portato da 1-> 0 o da 0-> a 1.

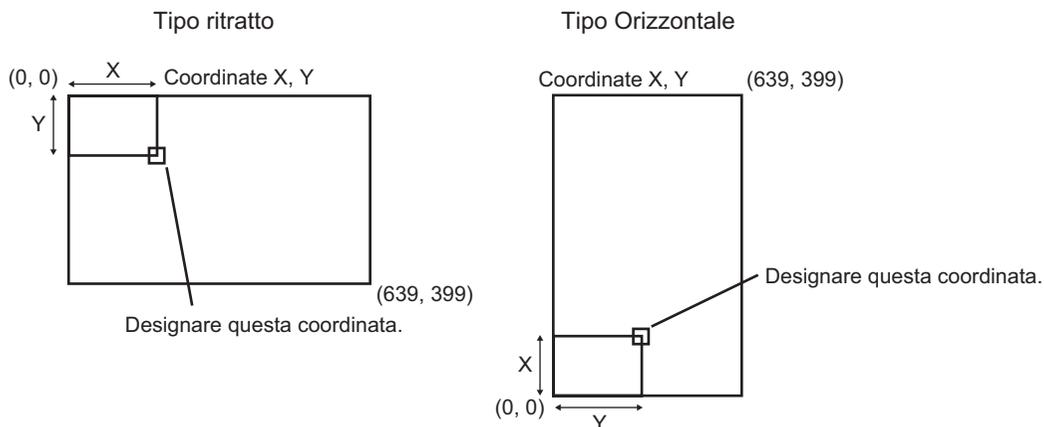
## 21.11.2 Disegna

Disegna	Riepilogo funzioni
	<p>Richiama schermo   " ■ Richiama schermo" (pagina 21-75)            Richiama lo schermo (schermo di base) con il numero designato.            Non si può utilizzare in uno script esteso.</p>
	<p>Cerchio   " ■ Linea" (pagina 21-77)            Disegna il cerchio designato.</p>
	<p>Dot   " ■ Dot" (pagina 21-77)            Disegna il dot designato.</p>
	<p>Linea   " ■ Linea" (pagina 21-77)            Disegna la linea designata.</p>
	<p>Rettangolo   " ■ Rettangolo" (pagina 21-78)            Disegna il rettangolo designato.</p>

## ■ Richiama schermo

Elemento	Descrizione
Riepilogo	<p>Questa funzione chiama un Elemento di libreria registrato. Lo schermo designato (Schermo di base) sarà richiamato presso le coordinate designate X,Y.            Non si può utilizzare in uno script esteso.</p>
Format	<p>b_call (Numero schermo, Coordinata X, Coordinata Y)</p> <div data-bbox="687 1238 954 1435" style="text-align: center;"> </div> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>Impostare la coordinata centrale dello schermo richiamato con le coordinate X e Y.</li> </ul>

### Posizione coordinata



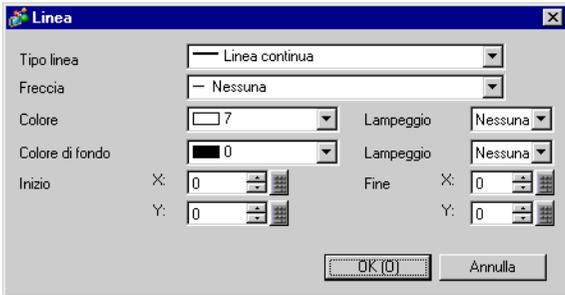
### ■ Cerchio

Elemento	Descrizione
Riepilogo	Disegnare un cerchio nel punto designato. Quando si seleziona la casella [Motivo], viene disegnato un cerchio pieno. Selezionare ed inserire il tipo di linea (o riempire il motivo quando lo si seleziona), gli attributi del colore, le coordinate centrali e il valore del raggio. Le coordinate centrali ed il raggio possono essere impostate indirettamente.
Format	<p>dsp_circle (Coordinata X, Coordinata Y, Raggio, Lampeggio colore Visualizzatore + Colore Visualizzatore, Lampeggio colore di sfondo + Colore di sfondo, Tipo di linea)</p> <div data-bbox="581 1083 1053 1377" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>• Quando si impostano il nero e il Lampeggio, il colore di fondo diventerà trasparente.</li> </ul>

## ■ Dot

Elemento	Descrizione
Riepilogo	Disegna un dot nel punto designato. Impostare le coordinate X, Y e il colore del Visualizzatore.
Format	<p>dsp_dot (Coordinata X, Coordinata Y, Lampeggio + Colore Visualizzatore)</p>  <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>• Quando si impostano il nero e il Lampeggio, il colore di fondo diventerà trasparente.</li> </ul>

## ■ Linea

Elemento	Descrizione
Riepilogo	Disegna una linea nel punto designato. Impostare il tipo di linea, gli attributi del colore e le coordinate iniziali e finali.
Format	<p>dsp_line (Punto iniziale coordinata X, Punto iniziale coordinata Y, Punto finale coordinata X, Punto finale coordinata Y, Lampeggio colore visualizzatore + Colore visualizzatore, Lampeggio colore di sfondo + Colore di sfondo, Tipo di linea e Freccia)</p>  <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>• Quando si impostano il nero e il Lampeggio, il colore di fondo diventerà trasparente.</li> </ul>

## ■ Rettangolo

Elemento	Descrizione
Riepilogo	<p>Disegna un rettangolo nel punto designato. Quando si seleziona la casella [Motivo], viene disegnato un rettangolo pieno.</p> <p>Selezionare ed inserire il tipo di linea (o riempire il motivo quando lo si seleziona), gli attributi del colore e le coordinate iniziali e finali.</p>
Format	<p>dsp_rectangle (Punto iniziale coordinata X, Punto iniziale coordinata Y, Punto finale coordinata X, Punto finale coordinata Y, Lampeggio colore visualizzatore + Colore visualizzatore, Lampeggio colore di sfondo + Colore di sfondo, Motivo e Tipo di linea)</p> <div data-bbox="487 531 1053 836" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> </div> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>• Quando si impostano il nero e il Lampeggio, il colore di sfondo diventerà trasparente.</li> </ul>

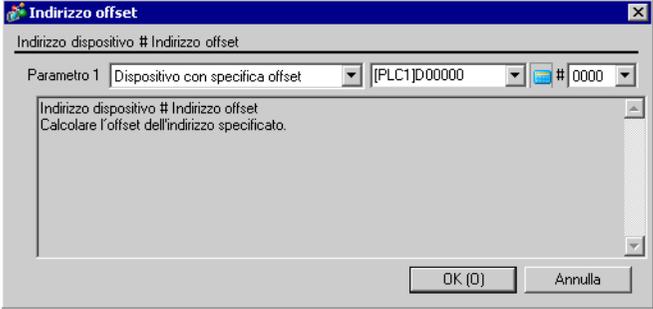
### IMPORTANTE

- Quando si usano colori nelle funzioni di disegno, impostare i codici colore da 0 a 255. Se si imposta da E1 a E12 e si salva lo script, si verificherà un errore.

## 21.11.3 Operazione memoria

Operazione memoria	Riepilogo funzioni
	<p>Indirizzo di specifica valore offset   " ■ Indirizzo offset" (pagina 21-80)            Imposta un offset indirizzo.</p>
	<p>Confronta memoria   " ■ Confronta memoria" (pagina 21-82)            Confronta due blocchi di dati nelle posizioni specificate (offset), e scrive il risultato del confronto nell'indirizzo di memorizzazione.</p>
	<p>Copia memoria   " ■ Copia memoria" (pagina 21-84)            Copia la memoria del dispositivo in una sola operazione.</p>
	<p>Copia memoria (Specificazione variabili)   " ■ Copia memoria (Variabile)" (pagina 21-88)            Copia la memoria del dispositivo in una sola operazione. Si possono modificare l'indirizzo di origine (copia da), quello di destinazione (copia a) e il numero di indirizzi.</p>
	<p>Ring memoria   " ■ Ring memoria" (pagina 21-89)            Sposta i dati in memoria in senso circolare per il numero designato di blocchi word.</p>
	<p>Cerca memoria   " ■ Cerca memoria" (pagina 21-91)            Esegue una ricerca dati in unità blocco, e restituisce (salva) i risultati di ricerca nell'indirizzo di memorizzazione specificato.</p>
	<p>Inizializza memoria   " ■ Inizializza memoria" (pagina 21-94)            Inizializza tutti i dispositivi in una volta sola.</p>
	<p>Inizializza memoria (Specificazione variabili)   " ■ Inizializza memoria (Variabile)" (pagina 21-95)            Inizializza tutti i dispositivi in una volta sola. Si possono modificare l'indirizzo in alto, i dati impostati e il numero di indirizzi.</p>
	<p>Sposta memoria   " ■ Sposta memoria" (pagina 21-96)            Sposta le unità di blocco verso l'alto.</p>

## ■ Indirizzo offset

Elemento	Descrizione																							
Riepilogo	Gli indirizzi offset possono essere designati. Solo gli indirizzi word temporanei possono essere designati per gli indirizzi di memorizzazione con valore offset.																							
Format	<p>[Indirizzo dispositivo] # [Indirizzo offset]</p>  <p>Intervalli di input costanti</p> <table border="1"> <thead> <tr> <th rowspan="2">Tipo di dati</th> <th colspan="2">Input costanti</th> </tr> <tr> <th>Min</th> <th>Max</th> </tr> </thead> <tbody> <tr> <td>Bin16</td> <td>0</td> <td>65535</td> </tr> <tr> <td>Bin32</td> <td>0</td> <td>4294967295</td> </tr> <tr> <td>Bin16+/-</td> <td>-32768</td> <td>32767</td> </tr> <tr> <td>Bin32+/-</td> <td>-2147483648</td> <td>2147483647</td> </tr> <tr> <td>BCD16</td> <td>0</td> <td>9999</td> </tr> <tr> <td>BCD32</td> <td>0</td> <td>99999999</td> </tr> </tbody> </table>	Tipo di dati	Input costanti		Min	Max	Bin16	0	65535	Bin32	0	4294967295	Bin16+/-	-32768	32767	Bin32+/-	-2147483648	2147483647	BCD16	0	9999	BCD32	0	99999999
Tipo di dati	Input costanti																							
	Min	Max																						
Bin16	0	65535																						
Bin32	0	4294967295																						
Bin16+/-	-32768	32767																						
Bin32+/-	-2147483648	2147483647																						
BCD16	0	9999																						
BCD32	0	99999999																						

### Espressione esempio 1:

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

Nell'esempio sopra, quando il valore di [t:0000] è 2, il valore memorizzato in D0102 sarà offset rispetto a D0200.

### Espressione esempio 2:

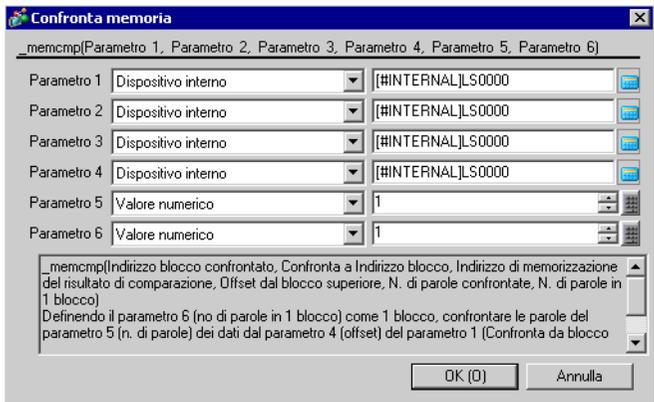
[w:[PLC1]D0100]#[t:0000]=30

Nell'esempio sopra, quando il valore di [t:0000] è 8, 30, sarà offset rispetto a D0108.

**IMPORTANTE**

- Gli indirizzi word utilizzati nell'indirizzo offset non vengono contati come indirizzi D-Script.
  - I dati provenienti da un dispositivo designati da un indirizzo offset non vengono continuamente estrapolati dal dispositivo collegato. Vengono letti quando si esegue il D-Script. Quando si verifica un errore durante l'unità di lettura, il valore di quest'ultima viene trattato come "0". Inoltre, il bit 12 del relé speciale intero LS2032 dell'unità di visualizzazione si accenderà. Quando la lettura dei dati viene completata normalmente, il bit 12 si porterà su OFF.
  - Se il risultato offset dell'indirizzo supera i 16 bit (Valore max: 65535), i bit fino a 15 saranno validi, e i bit dal 16 in poi saranno scartati.
  - Quando si definisce una variabile come l'indirizzo, specificare un array intero. Verificare che l'array intero sia abbastanza grande per ospitare tutti gli indirizzi consecutivi. Le operazioni non saranno valide se l'array non è abbastanza grande da memorizzare indirizzi consecutivi. Le operazioni, inoltre, non saranno valide se la variabile intera non è un array.
-

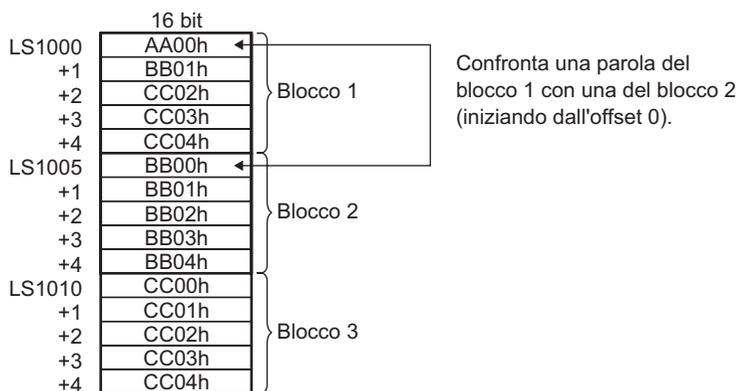
## ■ Confronta memoria

Elemento	Descrizione
Riepilogo	<p>Confronta due blocchi di dati nelle posizioni specificate (offset), e scrive il risultato del confronto nell'indirizzo di memorizzazione.</p> <p>I seguenti valori vengono memorizzati come risultato del confronto:            Quando i valori sono uguali: Quando i dati target hanno dimensioni superiori rispetto a quelli originali: Quando i dati target hanno dimensioni inferiori rispetto a quelli originali: Quando si verifica un errore, il valore di stato dell'errore viene scritto in LS9152.</p>
Format	<p>_memcmp ([Indirizzo blocco confrontato], [Confronta con l'indirizzo del blocco], [Indirizzo di memorizzazione dei risultati del confronto], Offset dall'inizio del blocco, Numero di word confrontate, N. di word nel blocco1)</p> <div style="text-align: center;">  </div> <p>Parametro 1: Dispositivo interno            Parametro 2: Dispositivo interno            Parametro 3: Dispositivo interno            Parametro 4: Valore numerico (da 0 a 639), Dispositivo interno, Variabile temporanea            Parametro 5: Valore numerico (da 1 a 640)            Parametro 6: Valore numerico (da 1 a 640)</p> <p>Dati da memorizzare            0: Corrispondenza            1: Confronto da &lt; Confronto a            2: Confronto da &gt; Confronto a</p>

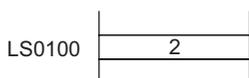
### Espressione esempio 1:

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

(Confronta una word del Blocco 1 con una del Blocco 2 (partendo da offset 0) e salva il risultato del confronto in LS0100)



Poiché il valore origine è inferiore al valore target, il risultato del confronto, “2”, viene memorizzato in LS0100.



**Espressione esempio 2:**

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

(Confronta una word del Blocco 1 con una del Blocco 3 (partendo da offset 2) e salva il risultato del confronto in LS0100).



Poiché i valori dei dati originali e dei dati target corrispondono, il risultato del confronto, “0”, viene memorizzato in LS0100



**Stato errore**

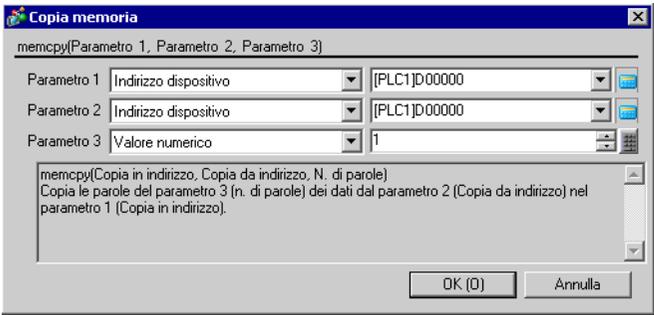
LS9152	Area LS

Nome funzione Editor	Area LS	Stato errore	Causa
_memcpy ( )	LS9152	0000h	Fase di completamento riuscita
		0001h	Errore parametro
		0003h	Errore lettura/scrittura

**IMPORTANTE**

- L'intervallo del dispositivo LS efficace in grado di essere specificato viene limitato all'area utente designata (da LS20 a LS2031 e da LS2096 a LS8191).
- Quando si specifica un valore per l'offset dalla cima del blocco, valore superiore al numero di word in un blocco, la funzione non sarà attivata.
- Quando il numero di word da confrontare è superiore a un blocco, la funzione non sarà attivata.

**■ Copia memoria**

Elemento	Descrizione
Riepilogo	Copia la memoria del dispositivo in una sola operazione. I dati per il numero di indirizzi saranno copiati negli indirizzi word di destinazione copia, iniziando dal primo indirizzo word dei dati di origine. Il numero di indirizzi in grado di essere utilizzato va da 1 a 640.
Format	memcpy ([Copia nell'indirizzo], [Copia dall'indirizzo], Word) 

**Espressione esempio:**

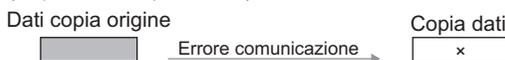
```
memcpy ([w:[PLC1]D0200], [w:[PLC1]D0100], 10)
```

Nell'esempio sopra, i dati vengono copiati da D0100-D0109 a D0200-D0209

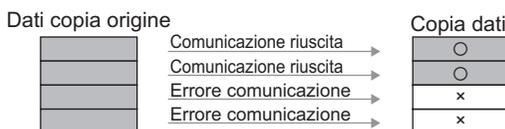
**IMPORTANTE**

- I dati di copia di origine saranno letti dal dispositivo collegato solo una volta, se necessario. Se si verifica un errore di comunicazione durante la lettura dei dati, il bit 12 del relé speciale interno dell'unità di visualizzazione, LS2032, si porterà su ON. Quando la lettura dei dati viene completata normalmente, il bit 12 si porterà su OFF.
- La lettura dei dati di copia di origine e la scrittura dei dati nella destinazione o target vengono eseguite in una sola operazione, oppure si eseguono dividendo i dati in vari elementi equivalenti al numero degli indirizzi usati per i dati di copia di origine. Se si verifica un errore di comunicazione durante la lettura dei dati, i risultati della copia dei dati varieranno come indicato di seguito, e differiranno se l'elaborazione dei dati avviene in una sola operazione o in diversi elementi. (Risultato dell'operazione di scrittura dati O: Scrittura completata, X: Impossibile scrivere)

(Copia in una operazione)



(Copia per i dati di divisione)



- Con l'aumentare del numero degli indirizzi, si impiegherà più tempo per scrivere i dati nel dispositivo PLC. Il tempo impiegato varierà dai 20 secondi a diversi minuti, in base al numero di indirizzi.
- Se i dati da scrivere superano la portata del dispositivo designato, si verificherà un errore di comunicazione. In questo caso, si dovrà spegnere e riaccendere l'unità GP per azzerare l'unità GP dopo l'errore.
- Quando i dati sono scritti nell'area LS Area con la funzione Copia memoria (memcpy), potranno essere scritti solo nell'area Utente. I dati non potranno essere scritti nell'area Dati sistema (da LS0000 a LS0019), nell'Area speciale (da LS2032 a LS2047), o nell'Area riservata (da LS2048 a LS2095). In ogni caso, i dati possono essere estrapolati da queste aree.

Continua

**IMPORTANTE**

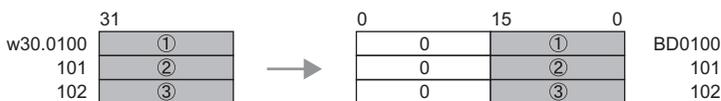
- Quando si usa il D-Script per copiare i dati del dispositivo a 32 bit in un dispositivo a 16 bit, e la lunghezza viene designata di 16 bit, solo i dati nei 16 bit inferiori saranno copiati.

Ad esempio, memcpy ([w:[PLC1]w30.0100], [w:[PLC1]BD0100], 3)



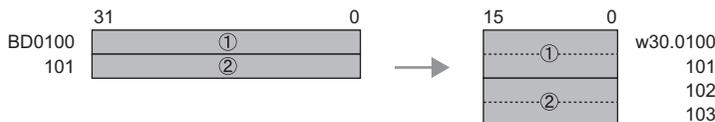
Inoltre, quando i dati del dispositivo a 16 bit sono copiati su un dispositivo a 32 bit, saranno copiati nei 16 bit di fondo, mentre per i 16 bit in cima sarà impostato il valore "0".

Ad esempio, memcpy ([w:[PLC1]BD0100], [w:[PLC1]w30.0100], 3)

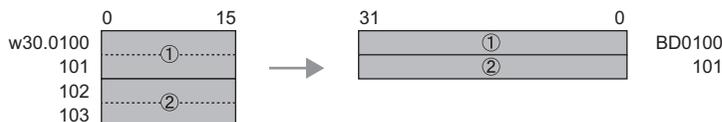


- Quando i dati del dispositivo a 32 bit vengono copiati in un dispositivo a 16 bit, o quando i dati del dispositivo a 16 bit vengono copiati in un dispositivo a 32 bit, se la lunghezza del bit del D-Script designata è 32, l'operazione di copia avverrà come segue. Quando uno dei dispositivi è a 32 bit e l'altro è un dispositivo a 16 bit, la funzione memcpy ( ) utilizzerà 16 bit come parametro di lunghezza dati.

Ad esempio, memcpy ([w:[PLC1]w30.0100], [w:[PLC1]BD0100], 4)



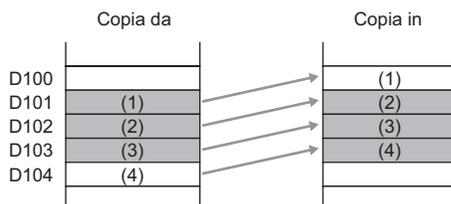
Ad esempio, memcpy ([w:[PLC1]BD0100], [w:[PLC1]w30.0100], 4)



- Se gli intervalli dei dati originali e di destinazione si sovrappongono, tutti i dati sovrapposti saranno riscritti come segue:

Ad esempio, quando si copia da D101-D104 a D100-D103

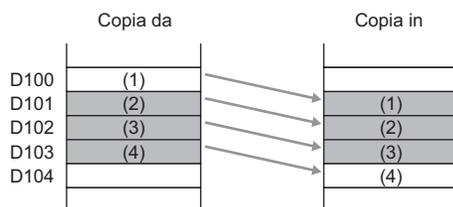
I dati saranno copiati in un indirizzo con un numero inferiore.



Continua

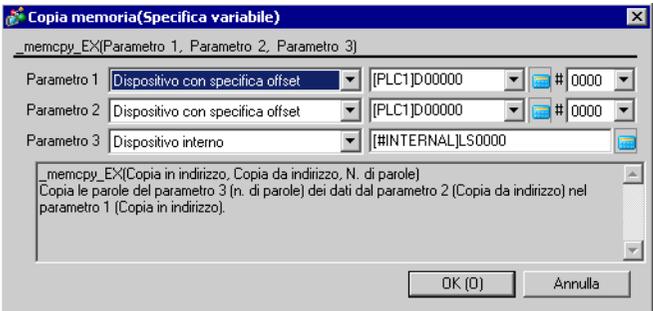
**IMPORTANTE**

Ad esempio, quando si copia da D100-D103 a D101-D104  
I dati saranno copiati in un indirizzo con un numero superiore.



- Benché la funzione di quest'esempio designi 2 indirizzi, questi non saranno contati come indirizzi D-Script.
  - Quando si usa un indirizzo di dispositivo per l'assegnazione, la comunicazione con il dispositivo/PLC causerà un leggero ritardo nell'assegnazione del valore.
-

## ■ Copia memoria (Variabile)

Elemento	Descrizione
Riepilogo	<p>Copia la memoria del dispositivo in una sola operazione. I dati degli indirizzi specificati con il Parametro 3 sono copiati dall'indirizzo word di origine specificato con il parametro 2 nell'indirizzo word di destinazione specificato con il Parametro 1.</p> <p>Il numero di indirizzi che possono essere utilizzati varia da 1 a 640. Con la funzione “_memcpy_EX”, l'indirizzo di origine, quello di destinazione e il numero di indirizzi potranno essere designati in modo indiretto.</p>
Format	<p>_memcpy_EX ([Copia nell'indirizzo], [Copia dall'indirizzo], Word)</p> <p>Parametro 1: Indirizzo del dispositivo + Indirizzo temporaneo            Parametro 2: Indirizzo del dispositivo + Indirizzo temporaneo            Parametro 3: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 3 va da 1 a 640).</p> 

### Espressione esempio:

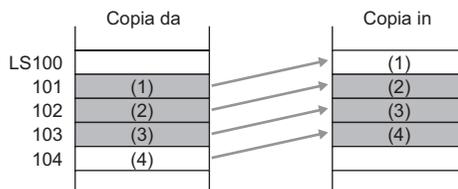
[t:0000]=10, [t:0001]=20

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

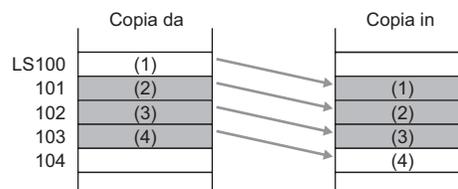
Nell'esempio sopra, cinque word di dati saranno estrapolate da D0120 e scritte da LS0110 a LS0114.

### IMPORTANTE

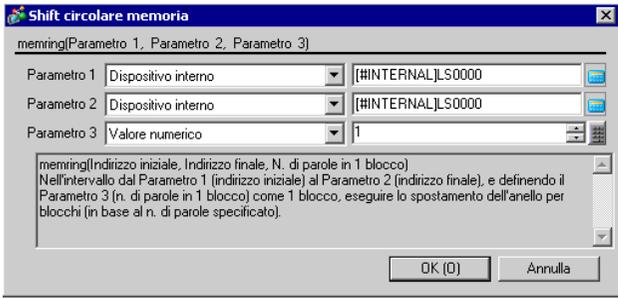
- Se gli intervalli dei dati originali e di destinazione si sovrappongono, tutti i dati sovrapposti saranno riscritti come segue:  
 Ad esempio, quando si copia da LS101-LS104 a LS101-LS103  
 I dati saranno copiati in un indirizzo con un numero inferiore.



Ad esempio, quando si copia da LS100-LS103 a LS101-LS104  
 I dati saranno copiati in un indirizzo con un numero superiore.



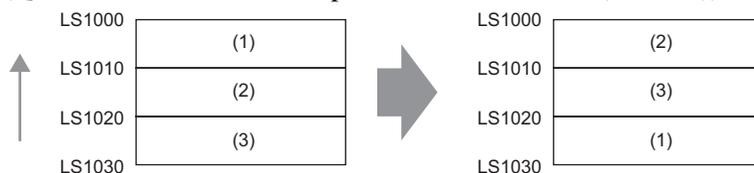
## ■ Ring memoria

Elemento	Descrizione
Riepilogo	Sposta ad anello i dati in memoria in blocchi. Esegue lo spostamento ad anello tra gli indirizzi iniziale e finale, in unità di blocco (per il numero specificato di word). Quando si verifica un errore, lo stato dell'errore viene scritto in LS9150.
Format	<p>memring ([Indirizzo iniziale], [Indirizzo finale], Word nel blocco 1)</p>  <p>Parametro 1: Dispositivo interno  Parametro 2: Dispositivo interno  Parametro 3: Valore numerico (da 1 a 640)</p> <p>Quando il parametro 1 è inferiore al Parametro 2 (<math>P1 &lt; P2</math>), i dati del blocco vengono spostati verso l'alto.  Quando il parametro 1 è superiore al Parametro 2 (<math>P1 &gt; P2</math>), i dati del blocco vengono spostati verso il basso.</p> <p><b>IMPORTANTE</b></p> <ul style="list-style-type: none"> <li>Verificare che l'indirizzo iniziale e quello finale siano impostati sullo stesso tipo di dispositivo (LS o USR).</li> </ul>

### Espressione esempio 1:

memring ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1030], 10)

(Quando il Parametro 1 è superiore al Parametro 2 ( $P1 < P2$ ))

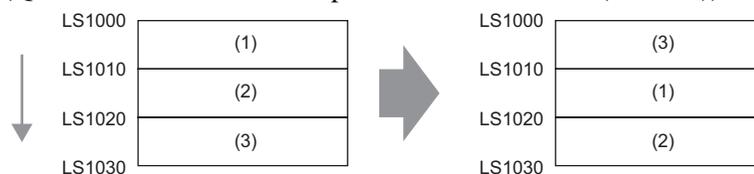


I dati si spostano verso l'alto in unità di blocchi da 10 word.

### Espressione esempio 2:

memring ([w:[#INTERNAL]LS1030], [w:[#INTERNAL]LS1000], 10)

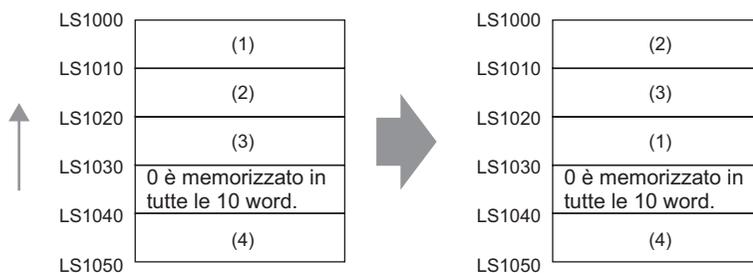
(Quando il Parametro 1 è superiore al Parametro 2 ( $P1 > P2$ ))



I dati si spostano verso il basso in unità di blocchi da 10 word.

**Espressione esempio 3:**

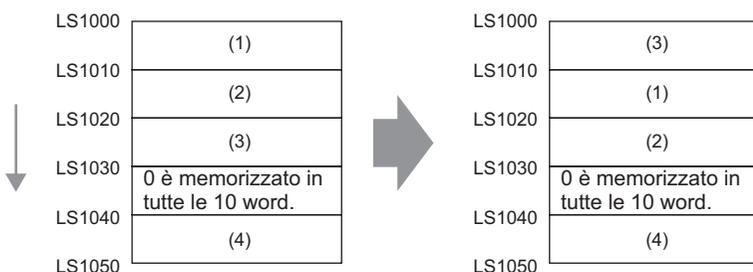
memring ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1050], 10)  
 (Quando l'intervallo contiene un blocco dove tutte le word sono uguali a "0".)



I dati vengono spostati verso l'alto solo in unità di blocchi da 10 word, dal blocco iniziale al blocco con dati "0". Eventuali dati dopo il blocco con dati "0" saranno ignorati.

**Espressione esempio 4:**

memring ([w:[#INTERNAL]LS1050], [w:[#INTERNAL]LS1000], 10)  
 (Quando un blocco con dati "0" esiste nell'intervallo.)



I dati vengono spostati verso il basso solo in unità di blocchi da 10 word, dal blocco iniziale al blocco con dati "0". Eventuali dati dopo il blocco con dati "0" saranno ignorati.

**Stato errore**

	Area LS
LS9150	

Nome funzione Editor	Area LS	Stato errore	Causa
memring ( )	LS9150	0000h	Fase di completamento riuscita
		0001h	Errore parametro
		0003h	Errore lettura/scrittura

**IMPORTANTE**

- Il tempo di elaborazione richiesto è proporzionale all'intervallo designato dagli indirizzi iniziale e finale. Più grande è l'intervallo designato più elevato sarà il tempo di elaborazione. La Parte non sarà aggiornata fino al termine dell'elaborazione.
- L'intervallo del dispositivo LS efficace in grado di essere specificato viene limitato all'area utente designata (da LS20 a LS2031 e da LS2096 a LS8191).

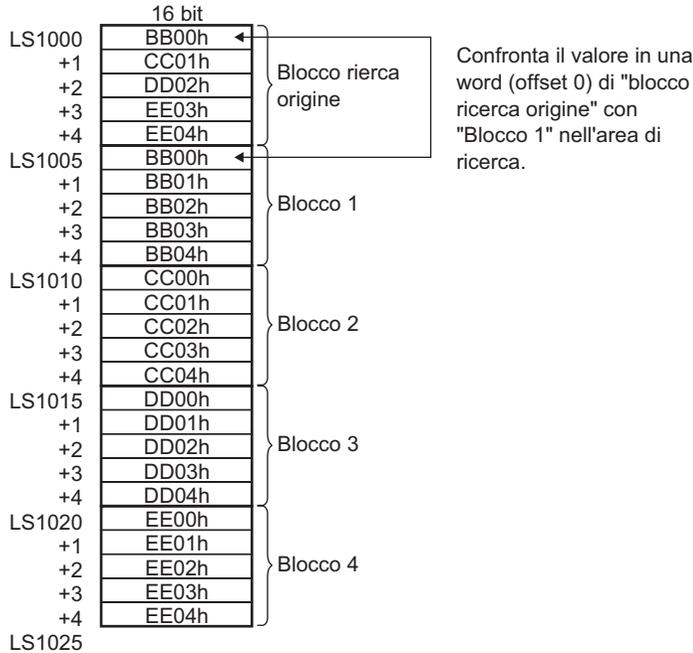
## ■ Cerca memoria

Elemento	Descrizione
Riepilogo	<p>Esegue una ricerca dati in unità blocco, iniziando dal primo elemento dell'intervallo specificato. Confronta blocchi di dati, iniziando dai blocchi specificati (offset) e restituisce (salva) il risultato della ricerca all'indirizzo di memorizzazione specificato. Quando si trova un blocco corrispondente, si salva il valore offset del blocco (1 o superiore). Quando non si trova alcun blocco corrispondente, si salverà "FFFFh". Quando si verifica un errore, il valore di stato dell'errore viene scritto in LS9153.</p>
Format	<p>_memsearch ([Indirizzo blocco ricercato], [Indirizzo di inizio ricerca], [Indirizzo di fine ricerca], [Indirizzo di memorizzazione dei risultati di ricerca], Offset dal blocco iniziale, Numero di word confrontate, Word in blocco 1)</p> <div data-bbox="454 633 1075 1045" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> </div> <p>Parametro 1: Dispositivo interno  Parametro 2: Dispositivo interno  Parametro 3: Dispositivo interno  Parametro 4: Dispositivo interno  Parametro 5: Valore numerico (da 0 a 639), Dispositivo interno, Variabile temporanea  Parametro 6: Valore numerico (da 1 a 640)  Parametro 7: Valore numerico (da 1 a 640)</p> <p>Dati da scrivere</p> <p>Quando esistono blocchi corrispondenti: Valore di offset del blocco ("1" o superiore)  Quando non esistono blocchi corrispondenti: "FFFFh"</p> <p><b>IMPORTANTE</b></p> <ul style="list-style-type: none"> <li>• Verificare che l'indirizzo di ricerca iniziale e quello finale siano impostati sullo stesso tipo di dispositivo (LS o USB). In ogni caso, l'[Indirizzo di blocco ricercato] e l'[Indirizzo di memorizzazione dei risultati di ricerca] possono essere impostati sul Dispositivo interno.</li> <li>• Verificare che il [Parametro 2] sia più piccolo del [Parametro 3]. Altrimenti, si verifica un errore.</li> </ul>

**Espressione esempio 1:**

```
_memsearch ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1005],
[w:[#INTERNAL]LS1025], [w:[#INTERNAL]LS0100], 0, 1, 5)
```

(Ricerca da LS1005 a LS1025 un blocco con lo stesso valore. Inizia dall'offset 0 del blocco di ricerca di origine, e memorizza il risultato in LS0100.)



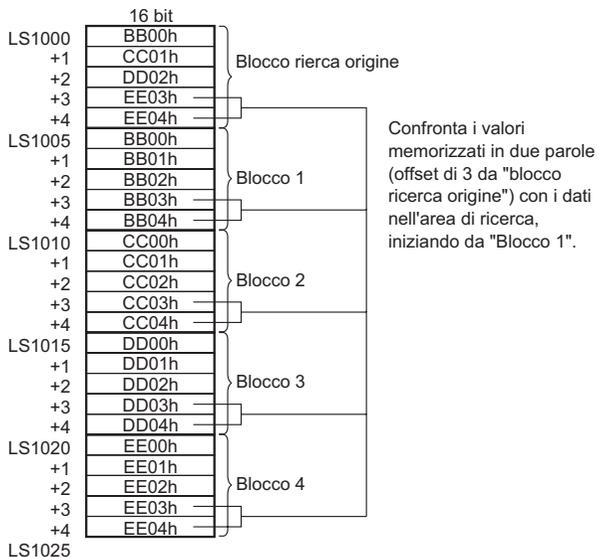
In questo caso, il valore di "Blocco 1" corrisponde al valore del "blocco di ricerca origine". Di conseguenza, il risultato di ricerca "1" viene memorizzato in LS0100.

LS0100	1
--------	---

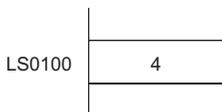
**Espressione esempio 2:**

```
_memsearch ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1005],
[w:[#INTERNAL]LS1025], [w:[#INTERNAL]LS0100], 3, 2, 5)
```

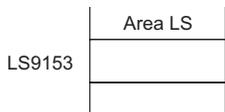
(Ricerca da LS1005 a LS1025 un blocco con lo stesso valore. Usa due word, iniziando dall'offset 3, e memorizza il risultato in LS0100.)



In questo caso, il valore di "Blocco 4" corrisponde al valore del "blocco di ricerca origine". Di conseguenza, il risultato di ricerca "4" viene memorizzato in LS0100.



**Stato errore**

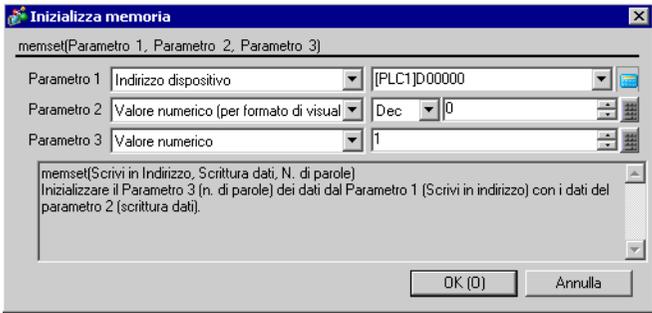


Nome funzione Editor	Area LS	Stato errore	Causa
_memsearch ( )	LS9153	0000h	Fase di completamento riuscita
		0001h	Errore parametro
		0003h	Errore lettura/scrittura

**IMPORTANTE**

- Il tempo di elaborazione richiesto è proporzionale all'intervallo designato dagli indirizzi iniziale e finale. Più grande è l'intervallo designato più elevato sarà il tempo di elaborazione. La Parte non sarà aggiornata fino al termine dell'elaborazione.
- L'intervallo del dispositivo LS efficace in grado di essere specificato viene limitato all'area utente designata (da LS20 a LS2031 e da LS2096 a LS8191).

## ■ Inizializza memoria

Elemento	Descrizione
Riepilogo	Inizializza tutti i dispositivi in una volta sola. L'impostazione dei dati per il numero di indirizzi viene estratta dall'Indirizzo word stabilito. L'intervallo valido per il numero di indirizzi va da 1 a 640.
Format	memset ([Indirizzo Scrivi-in], Dati scrittura, word) 

### Espressione esempio:

```
memset ([w:[PLC 1]D0100], 0, 10)
```

Nell'esempio sopra, "0" viene impostato per gli indirizzi da D0100 a D0109.

### IMPORTANTE

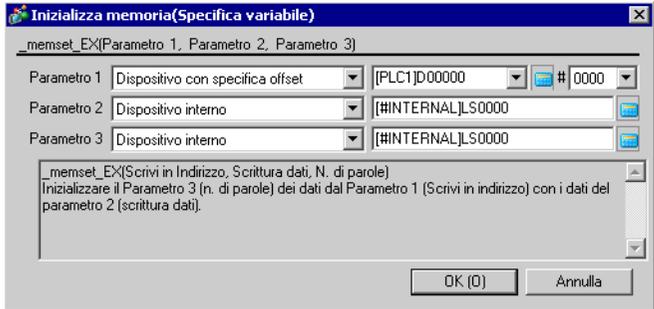
- Con l'aumentare del numero degli indirizzi, si impiegherà più tempo per scrivere i dati nel dispositivo PLC. Il tempo impiegato varierà dai 20 secondi a diversi minuti, in base al numero di indirizzi.
- Se i dati da scrivere superano la portata del dispositivo designato, si verificherà un errore di comunicazione. In questo caso, si dovrà spegnere e riaccendere l'unità GP per azzerare l'unità GP dopo l'errore.
- Benché questa funzione designi vari indirizzi, questi non saranno contati come indirizzi D-Script.
- Quando i dati sono scritti nell'area LS Area con la funzione Azzerare memoria (memset), potranno essere scritti solo nell'area Utente. I dati non potranno essere scritti nell'area Dati sistema (da LS0000 a LS0019), nell'Area speciale (da LS2032 a LS2047), o nell'Area riservata (da LS2048 a LS2095).
- Quando si usano indirizzi del dispositivo per l'operazione di assegnazione, i valori di scrittura non saranno assegnati immediatamente, a causa del tempo di trasmissione dall'unità GP al dispositivo PLC.

Ad esempio:

```
memset ([w:[PLC1]D0100], 0, 10) //Inizializza D100 da D109 a 0
[w:[PLC1]D200]=[w:[PLC1]D100] //Sostituisce D100 in D200
```

In questo caso, il valore 0 scritto a D100 come risultato dell'operazione non è stato ancora assegnato a D200.

## ■ Inizializza memoria(Variabile)

Elemento	Descrizione
Riepilogo	Inizializza tutti i dispositivi in una volta sola. I dati Impostati specificati con il Parametro 2 sono impostati dall'Indirizzo word Imposta specificato nel Parametro 1 agli indirizzi specificati con il Parametro 3. L'intervallo valido per il numero di indirizzi va da 1 a 640. L'indirizzo Scrivi-in, Dati scrittura, e il numero di indirizzi possono essere designati indirettamente.
Format	<p><code>_memset_EX</code> ([Indirizzo Scrivi-in], Dati scrittura, word)</p>  <p>Parametro 1: Indirizzo del dispositivo + Indirizzo temporaneo  Parametro 2: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 2 va da 0 a 65535 per Dec, e da 0 a FFFF per Esad.)  Parametro 3: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 3 va da 1 a 640).</p>

### Espressione esempio:

[t:0000]=10

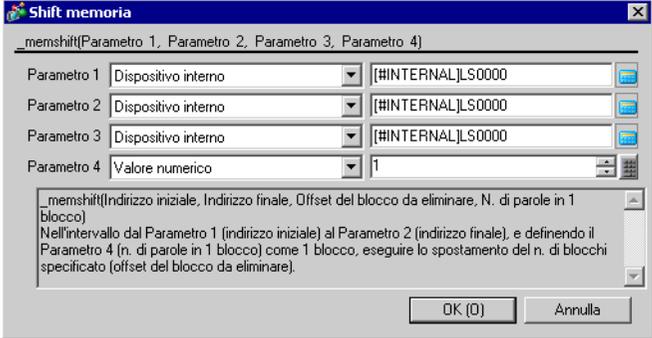
[ w:[#INTERNAL]LS0050]=0

[ w:[#INTERNAL]LS0050]=5

`_memset_EX` ([w:[#INTERNAL]LS0100]#[t:0000], [w:[#INTERNAL]LS0050],  
[w:[#INTERNAL]LS0051])

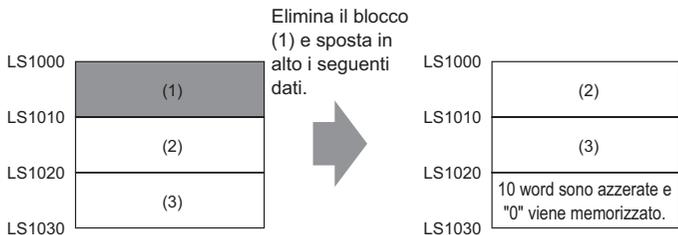
Nell'esempio sopra, "0" sarà scritto nelle cinque word da LS0100 a LS0114.

## ■ Sposta memoria

Elemento	Descrizione
Riepilogo	Elimina il blocco specificato e sposta i blocchi di dati seguenti verso l'alto. Il blocco da eliminare viene designato utilizzando un offset. Quando si verifica un errore, lo stato dell'errore viene scritto in LS9151.
Format	<p><code>_memshift([Indirizzo iniziale], [Indirizzo finale], Offset del blocco da eliminare, Word nel blocco 1)</code></p>  <p>Parametro 1: Dispositivo interno  Parametro 2: Dispositivo interno  Parametro 3: Valore numerico (da 0 a 65535), Dispositivo interno, Variabile temporanea  Parametro 4: Valore numerico (da 1 a 640)</p> <p><b>IMPORTANTE</b></p> <ul style="list-style-type: none"> <li>• Verificare che l'indirizzo iniziale e quello finale siano impostati sullo stesso tipo di dispositivo (LS o USR).</li> <li>• Verificare che il [Parametro 1] sia più piccolo del [Parametro 2]. Altrimenti, si verifica un errore.</li> </ul>

**Espressione esempio 1:**

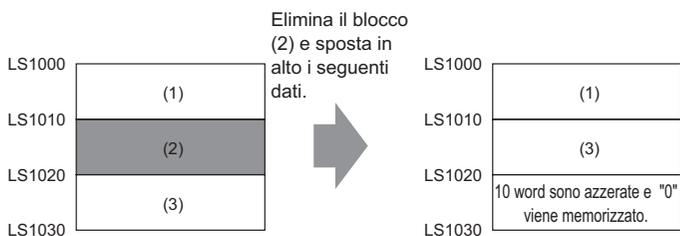
`_memshift ([w:[#INTERNAL]LS 1000], [w:[#INTERNAL]LS1030], 1, 10)`



I dati si spostano verso l'alto in unità di blocco (1 blocco = 10 word), e l'ultimo blocco (10 word) viene azzerato.

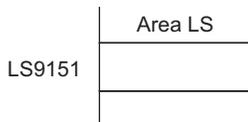
**Espressione esempio 2:**

`_memshift ([w:[#INTERNAL]LS 1000], [w:[#INTERNAL]LS1030], 2, 10)`



I dati si muovono verso l'alto in unità di blocco (1 blocco = 10 word) iniziando dalla posizione offset 2, e l'ultimo blocco (10 word) viene azzerato.

**Stato errore**

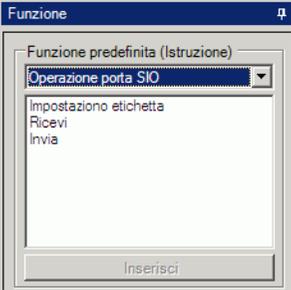


Nome funzione Editor	Area LS	Stato errore	Causa
_memshift ( )	LS9151	0000h	Fase di completamento riuscita
		0001h	Errore parametro
		0003h	Errore lettura/scrittura

**IMPORTANTE**

- Il tempo di elaborazione richiesto è proporzionale all'intervallo designato dagli indirizzi iniziale e finale. Più grande è l'intervallo designato più elevato sarà il tempo di elaborazione. La Parte non sarà aggiornata fino al termine dell'elaborazione.
- Quando un valore che supera l'intervallo specificato per gli indirizzi iniziale e finale viene designato come offset del blocco da eliminare, la funzione non opererà correttamente.
- L'intervallo del dispositivo LS efficace in grado di essere specificato viene limitato all'area utente designata (da LS20 a LS2031 e da LS2096 a LS8191).

## 21.11.4 Operazione porta SIO

Operazione porta SIO	Riepilogo funzioni
	<p>Impostazioni etichette   " ■ Impostazioni etichette" (pagina 21-100)            Impostato dalle Funzioni Controllo, Stato, Ricevi conteggio dati, Funzione Ricevi, e Funzione Invia.</p>
	<p>Ricevi   " ■ Ricevi" (pagina 21-102)            Legge i dati ricevuti dalla porta seriale designata (COM1 o COM2).</p>
	<p>Invia   " ■ Invia" (pagina 21-103)            Scrive alla porta seriale designata (COM1 o COM2).</p>
	<p>Ricezione estesa   " ■ Ricezione estesa" (pagina 21-104)            Legge i dati ricevuti dalla porta seriale designata (COM1 o COM2).            Può essere utilizzata solo in uno Script esteso.</p>
	<p>Invio esteso   " ■ Invio esteso" (pagina 21-105)            Scrive alla porta seriale designata (COM1 o COM2).            Può essere utilizzata solo in uno Script esteso.</p>
	<p>Funzione di ricezione in standby   " ■ Funzione ricezione Standby" (pagina 21-106)            Rimane in modalità ricezione in standby fino a quando riceve il testo specificato.            Può essere utilizzata solo in uno Script esteso.</p>
	<p>Funzione Standby   " ■ Funzione di standby" (pagina 21-107)            Il sistema attende per il periodo specificato.            Può essere utilizzata solo in uno Script esteso.</p>

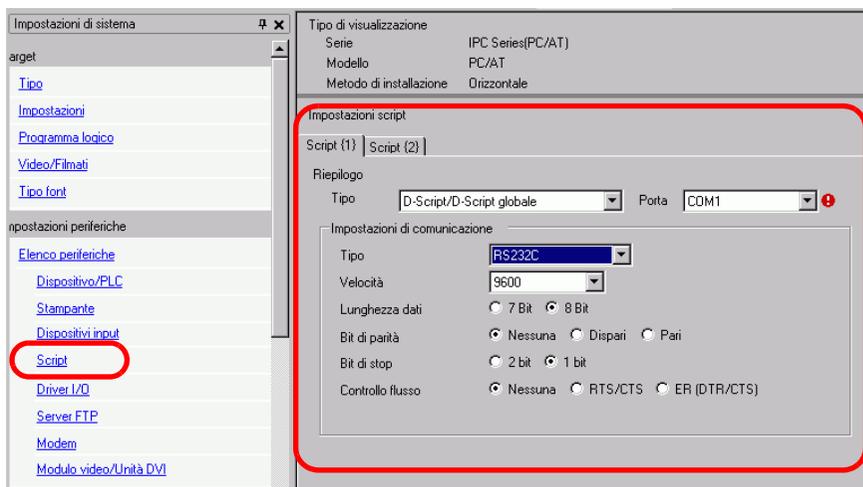
**IMPORTANTE**

- Impostazioni etichette, Invia e Ricevi possono essere facilmente incluse in un D-Script/D-Script globale.
- Per comunicare con i D-Script/ D-Script globali, impostare i seguenti script. Se le impostazioni di script non sono designate, gli script non potranno essere eseguiti.

[Procedura I/O di un D-Script/ D-Script globale]

Nella finestra [Impostazioni di sistema] fare clic su [Script].

Impostare il [Tipo] su [D-Script/D-Script globale].



Esistono due etichette per le impostazioni di script. L'esempio riportato sopra usa [Script1].

Impostare la [Porta] su COM1 o COM2, e stabilire le [Impostazioni di comunicazione] in modo che corrispondano al SIO esteso.

- Quando si crea un programma di comunicazioni con funzionalità più avanzate di quelle dell'operazione porta SIO, si raccomanda di utilizzare uno [Script esteso]. Per alcuni esempi dell'uso degli script estesi, vedere "21.5 Comunicazione con periferiche non supportate" (pagina 21-20)

## ■ Impostazioni etichette

### Controllo

Elemento	Descrizione
Riepilogo	La variabile Controllo viene utilizzata per azzerare il buffer Invia, il buffer Ricevi e lo stato di errore. Questa variabile è di sola scrittura.
Format	Quando si designa il bit: [c:EXT_SIO_CTRL**]**: da 00 a 15) Quando si designa la word: [c:EXT_SIO_CTRL]

### ◆ Espressione esempio:

Quando si designa il bit: [c:EXT\_SIO\_CTRL00] = 1

Quando si designa la word: [c:EXT\_SIO\_CTRL] = 0x0007

### ◆ EXT\_SIO\_CTRL

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

Bit	Contenuto
15	Riservato
14	
13	
12	
11	
10	
9	
8	
7	
6	
5	
4	
3	1: Azzerà timeout ricezione
2	1: Azzerà errore
1	1: Azzerà buffer ricezione
0	1: Azzerà buffer invio

#### NOTA

- Quando si seleziona una word e due o più bit sono impostati simultaneamente, l'elaborazione verrà eseguita nell'ordine indicato di seguito: Azzerà errore → Azzerà buffer di ricezione → Azzerà buffer di invio

### Stato

Elemento	Descrizione
Riepilogo	Lo stato include le informazioni riportate di seguito. Questa variabile Stato è di sola scrittura.
Format	Quando si designa il bit: [s:EXT_SIO_STAT**] (** : da 00 a 15) Quando si designa la word: [s:EXT_SIO_STAT]

### ◆ Espressione esempio:

Quando si designa il bit: if ([s:EXT\_SIO\_STAT 00] == 1)

Quando si designa la word: se (([s:EXT\_SIO\_STAT] & 0x0001) <> 0)

◆ **Contenuto di EXT\_SIO\_STAT**

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

Bit	Contenuto
15	0: Nessun D-Script/D-Script globale 1: D-Script/D-Script globale esiste
14	0: Nessuno script esteso 1: Lo script esteso esiste
13	Riservato
12	
11	
10	
9	
8	
7	
6	0: Normale
5	
4	0: Normale
3	1: Timeout ricezione 0: Normale
2	1: Receive error 0: Nessun dato ricevuto
1	1: Esiste Ricevi dati 0: Normale
0	1: Invia errore 0: Esistono dati nel buffer Invia

**NOTA**

- I bit riservati potrebbero essere assegnati in futuro. Pertanto, assicurarsi di selezionare solo i bit necessari.
- Esistono due tipi di errore di trasmissione: l'errore di timeout trasmissione e l'errore di buffer pieno trasmissione. Quando si verifica ciascuno di questi errori, il bit di trasmissione si porterà su ON. Il periodo di timeout della trasmissione è di cinque secondi.
- Esistono quattro tipi di errore di ricezione: errore di parità, errore di sovrapposizione, errore di framing ed errore di sovraccarico. Quando si verifica uno di questi quattro errori, il bit dell'errore di ricezione si porterà su ON.
- Se si rileva un errore di trasmissione, i dati Invia rimarranno nel buffer di trasmissione. Se non si riesce a rilevare un errore di trasmissione, i dati Invia saranno trasmessi dal buffer di trasmissione.
- Quando si usa l'interfaccia seriale COM2, che è RS-422, il segnale CS (CTS) non può essere rilevato. Pertanto, non si potrà individuare un cavo staccato.

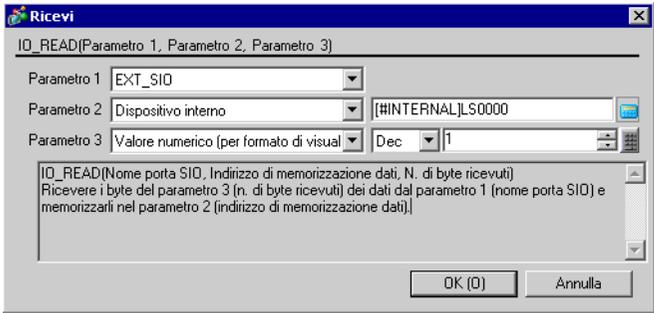
**Dimensioni dati ricevuti**

Elemento	Descrizione
Riepilogo	Mostra la quantità di dati (numero di byte) ricevuta in quel momento. Le dimensioni dei dati ricevuti costituiscono una funzione di sola lettura.
Format	[r:EXT_SIO_RECV]

**IMPORTANTE**

- Nome dell'etichetta Numero di dati ricevuti (numero di byte)  
 Con GP-PRO/PB III Vers. 6.0 e precedenti, il Nome etichetta designato per le dimensioni dei dati Ricevi è [r: EXT\_SIO\_RCV]. In ogni caso, non è necessario rivedere la descrizione, perché la funzione sarà la stessa, qualunque espressione [r: EXT\_SIO\_RCV] o [r: EXT\_SIO\_RECV] sia selezionata.

■ **Ricevi**

Elemento	Descrizione
Riepilogo	Scrivere l'istruzione come segue, quando si estrapolano i dati ricevuti dal SIO esteso.
Format	<p>IO_READ ([p:EXT_SIO], Indirizzo di memorizzazione dati, Numero di byte Ricevi)</p>  <p>Parametro 1: EXT_SIO                  Parametro 2: Dispositivo interno                  Parametro 3: Valore numerico</p>

**Espressione esempio:**

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

Nell'esempio sopra, il numero di byte ricevuti viene memorizzato in LS0100. 10 byte di dati vengono memorizzati iniziando da LS0101. L'immagine seguente mostra i dati ricevuti memorizzati.

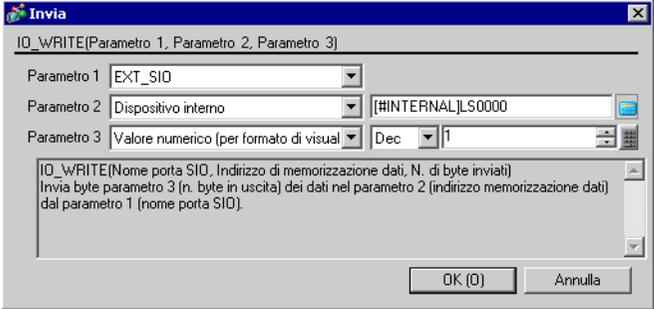
**NOTA**

- Il numero massimo di byte di trasferimento durante la ricezione dei dati è 2011. I dati vengono scritti in ciascun indirizzo word in unità di 1 byte.

LS0100	Dimensioni dati ricevuti	... 10 bytes
LS0101	00	Byte 1
LS0102	00	Byte 2
LS0103	00	Byte 3
LS0104	00	Byte 4
LS0105	00	Byte 5
LS0106	00	Byte 6
LS0107	00	Byte 7
LS0108	00	Byte 8
LS0109	00	Byte 9
LS0110	00	Byte 10

Metodo memorizzazione dati ricevuti

## ■ Invia

Elemento	Descrizione
Riepilogo	Scrivere l'istruzione come segue, quando si scrivono i dati ricevuti dal SIO esteso.
Format	<p>IO_WRITE ([p:EXT_SIO], Indirizzo di memorizzazione dati, Numero di byte Invia)</p>  <p>Parametro 1: EXT_SIO          Parametro 2: Dispositivo interno          Parametro 3: Valore numerico</p>

### Espressione esempio:

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

Nell'esempio riportato sopra vengono inviati 10 byte di dati, iniziando da LS0100.

L'immagine seguente mostra i dati inviati memorizzati.

#### NOTA

- Il massimo numero di byte di trasferimento quando si ricevono dati è 2012.
- Come dispositivo LS per il buffer Invia, scrivere i dati in byte singoli per ciascuno degli indirizzi word.

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

Invia metodo di memorizzazione dati

## ■ Ricezione estesa

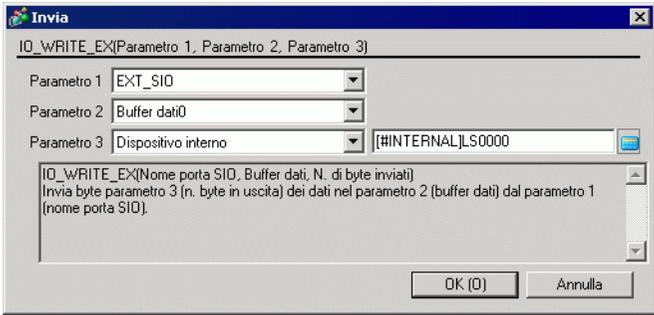
Elemento	Descrizione
Riepilogo	<p>Riceve dati delle dimensioni indicate in Dimensioni dati ricevuti (byte) dal SIO esteso e li memorizza nel buffer dati. Il numero di byte specificati con il Parametro 3 viene ricevuto dal SIO esteso e memorizzato nel buffer dati specificato con il Parametro 2.</p> <p>Può essere utilizzata solo in uno Script esteso.</p>
Format	<p>IO_READ_EX ([p:EXT_SIO], Buffer dati, Numero di byte Ricevi)</p> <div data-bbox="467 479 1122 794" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Parametro 1: [p:EXT_SIO]  Parametro 2: Buffer dati  Parametro 3: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 3 va da 1 a 1024).</p>

### Espressione esempio:

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

Nell'esempio riportato sopra, 10 byte dei dati ricevuti dal SIO esteso sono memorizzati in "databuf1".

## ■ Invio esteso

Elemento	Descrizione
Riepilogo	Invia i dati nel buffer dati con il SIO esteso secondo le dimensioni del numero di byte Invia. I contenuti del buffer dati specificati con il Parametro 2 vengono inviati dal SIO esteso secondo la lunghezza specificata con il Parametro 3. Può essere utilizzata solo in uno Script esteso.
Format	<p>IO_WRITE_EX ([p:EXT_SIO], Buffer dati, Numero di byte Invia)</p>  <p>Parametro 1: [p:EXT_SIO]  Parametro 2: Buffer dati  Parametro 3: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 3 va da 1 a 1024).</p>

### Espressione esempio:

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

Nell'esempio riportato sopra vengono inviati 10 byte di dati in "databuf0" dal SIO esteso.

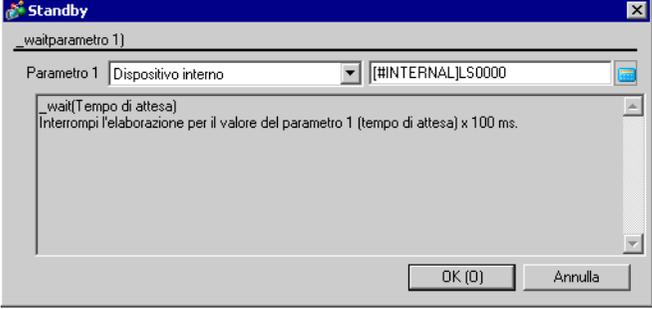
## ■ Funzione ricezione Standby

Elemento	Descrizione
Riepilogo	<p>Rimane in modalità ricezione in standby fino a quando riceve il testo specificato. Dopo la scadenza del periodo di timeout, il bit 4 (Ricevi errore di time-out ) di stato [s: EXT_SIO_STAT] viene impostato. La durata del timeout può essere stabilita in incrementi di 100 ms.</p> <p>Il sistema si trova in modalità di ricezione in standby fino a quando riceve la stringa di caratteri o il codice dei caratteri specificato nel Parametro 2. Configurare la durata del timeout nel Parametro 3.</p> <p>Può essere utilizzata solo in uno Script esteso.</p>
Format	<p>IO_READ_WAIT([p:EXT_SIO], Testo, Timeout)</p> <div data-bbox="458 581 1112 890" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Parametro 1: [p:EXT_SIO]  Parametro 2: Valore numerico, Testo, Buffer dati  Parametro 3: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 3 va da 1 a 600).</p>

### IMPORTANTE

- I dati ricevuti non potranno essere utilizzati fino alla ricezione del testo specificato. (In caso contrario, i dati verranno abbandonati).
- Si possono specificare fino a 128 caratteri (byte). Nota: l'operazione di ricezione in standby non può essere eseguita correttamente quando si specificano stringhe che superano il limite.

## ■ Funzione di standby

Elemento	Descrizione
Riepilogo	Il sistema attende per il periodo specificato. La durata può essere configurata in incrementi di 100 ms. Può essere utilizzata solo in uno Script esteso.
Format	<p><code>_wait(Tempo di attesa)</code></p>  <p>Parametro 1: , Dispositivo interno, Indirizzo temporaneo, Valore numerico (L'intervallo valido per il Parametro 1 va da 1 a 600).</p>

### Espressione esempio:

`_wait (10)`

Nell'esempio sopra, il sistema attende per un secondo.

## 21.11.5 Operazione file CF/Operazione file USB

Operazione file CF	Riepilogo funzioni
 	<p>Impostazioni etichette</p> <p>☞ " ■ Impostazioni etichette" (pagina 21-109)</p> <p>Le etichette vengono impostate dal numero di file elencati, dal numero di byte letti e dallo stato di errori nella CF-car/ memoria USB.</p>
	<p>Scrivi file</p> <p>☞ " ■ Scrivi file" (pagina 21-119)</p> <p>Si può selezionare una qualsiasi tra le tre modalità:</p>
	<p>Cambia nome file</p> <p>☞ " ■ Cambia nome file" (pagina 21-124)</p> <p>Modifica il nome del file.</p>
	<p>Leggi file CSV</p> <p>☞ " ■ Leggi file CSV" (pagina 21-126)</p> <p>Legge i dati in unità di celle da un file CSV e li scrive in un indirizzo word</p>
	<p>Leggi file</p> <p>☞ " ■ Leggi file" (pagina 21-129)</p> <p>Legge il numero specificato di byte di dati nel file dopo l'offset specificato, e lo scrive nell'indirizzo di destinazione.</p>
	<p>Elenco file output</p> <p>☞ " ■ Elenco file output" (pagina 21-132)</p> <p>L'elenco di file esistenti nella cartella specificata viene scritto nel dispositivo interno.</p>
	<p>Elimina file</p> <p>☞ " ■ Elimina file" (pagina 21-134)</p> <p>Elimina il file.</p>

## ■ Impostazioni etichette

I seguenti sono i possibili valori di stato per la CF-card/memoria USB.

Nome stato	Nome etichetta	Descrizione
File elencati	[s:CF_FILELIST_NUM] [s:USB_FILELIST_NUM]	Memorizza il numero di file effettivamente elencati quando viene eseguita la funzione di Output elenco file “_CF_dir ( )” o “_USB_read ( )” .
Numero di byte letti	[s:CF_READ_NUM] [s:USB_READ_NUM]	Memorizza il numero di byte che può essere estrapolato quando viene eseguita la funzione di Lettura file “_CF_read ( )” o “_USB_read ( )” .
Errore CF-card/ Errore memoria USB	[s:CF_ERR_STAT] [s:USB_ERR_STAT]	Memorizza lo stato di errore generato quando si accede alla CF-card o alla memoria USB.

### File elencati

Quando la funzione di Output elenco file “\_CF\_dir ( )” viene eseguita, il numero di elenchi file attualmente scritti nell'area LS viene memorizzato in “File elencati [s:CF\_FILELIST\_NUM]”/[s:USB\_FILELIST\_NUM]”.

### ◆ Esempio uso

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

```

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

Quando si ottiene un elenco di file contenente i 10 file, e la cartella specifica ne contiene solo cinque, il numero “5” sarà memorizzato in [s:CF\_FILELIST\_NUM].

#### **IMPORTANTE**

- Quando nessun file viene scritto, il numero totale dei file contenuti nella cartella specificata viene scritto in [s:CF\_FILELIST\_NUM].

### Numero di byte letti

Quando la funzione Lettura file “\_CF\_read ( )” or “\_USB\_read ( )” viene eseguita, il numero di byte effettivamente estrapolati viene memorizzato in “Byte estrapolati [s:CF\_READ\_NUM]”/[s:USB\_READ\_NUM].

### ◆ Esempio uso

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

Quando viene effettuato un tentativo di leggere 16 byte, ma si riesce a leggerne solo 12, “12” verrà memorizzato in [s:CF\_READ\_NUM].

**Stato errore CF-card/errore memoria USB**

Memorizza lo stato di errore generato quando si accede alla CF-card o alla memoria USB.

Posizione del bit	Nome errore	Descrizione
15	Riservato	Riservato
14		
13		
12		
11		
10		
9		
8		
7		
6	Errore di rinomina file	<ul style="list-style-type: none"> <li>• La CF-card/memoria USB viene rimossa durante l'esecuzione.</li> <li>• Il file specificato non esiste.</li> </ul>
5	Errore di eliminazione file	<ul style="list-style-type: none"> <li>• La CF-card/memoria USB viene rimossa durante l'esecuzione.</li> <li>• Il file specificato non esiste.</li> <li>• Si è tentato di eliminare un file con un attributo di sola lettura.</li> </ul>
4	Errore di scrittura file	<ul style="list-style-type: none"> <li>• La CF-card/memoria USB viene rimossa durante l'esecuzione.</li> <li>• Spazio libero della capacità della CF-card/memoria USB superato.</li> <li>• Si è tentato di scrivere dati in un file con un attributo di sola lettura.</li> <li>• Si è tentato di "sovrascrivere" un file che non esiste.</li> </ul>
3	Errore di lettura file	<ul style="list-style-type: none"> <li>• La CF-card/memoria USB viene rimossa durante l'esecuzione.</li> <li>• Il file specificato non esiste.</li> </ul>
2	Errore nell'elenco file	<ul style="list-style-type: none"> <li>• La CF-card/memoria USB viene rimossa durante l'esecuzione.</li> <li>• La cartella specificata non esiste.</li> </ul>
1	Errore CF-card/memoria USB	<ul style="list-style-type: none"> <li>• La CF-card/memoria USB non è valida.</li> <li>• Il supporto inserito non è una CF-card.</li> </ul>
0	Errore CF-card/memoria USB	<ul style="list-style-type: none"> <li>• La CF-card/memoria USB non è inserita.</li> <li>• Il portello di copertura è aperto.</li> </ul>

- Anche quando si verifica un errore di memoria nella CF-card/memoria USB, l'elaborazione continuerà. Assicurarsi di scrivere lo script, in modo da fargli controllare gli errori quando si usano le funzioni dell'operazione file di una CF-card/memoria USB. Ad esempio:

```
_CF_dir ("\\DATA\\*.*", [w:[#INTERNAL]LS0100], 2, 1) Emette un elenco di file.
if([s:CF_ERR_STAT02] <> 0) // Controlla lo stato dell'errore.
{
    set ([b:[#INTERNAL]LS005000]) // Imposta l'indirizzo di bit per la
                                visualizzazione di errori.
}
endif
```

◆ **CF Card/memoria USB - Area di memorizzazione stato dei dettagli degli errori**

Ogni bit sarà impostato quando si verificherà un errore. Si possono verificare i fattori che hanno condotto all'errore impostando Stato dettagli. In ciascuna funzione, lo Stato dettagli è memorizzato da LS9132 a LS9137 per Area sistema esteso (da LS9138 a LS9143 per la memoria USB). Queste aree sono di sola lettura.

Area LS		LS Area	
LS0000		LS0000	
:		:	
LS9132	Stato dell'operazione di elenco CF-card	LS9138	USB List Status
LS9133	Stato dell'operazione di lettura CF-card	LS9139	USB Read Status
LS9134	Stato dell'operazione di scrittura CF-card	LS9140	USB Write Status
LS9135	Stato dell'operazione di eliminazione CF-card	LS9141	USB Delete Status
LS9136	Stato dell'operazione di rinomina CF-card	LS9142	USB Rename Status
LS9137	Stato della lettura CSV	LS9143	USB CSV Read Status
:		:	
LS9999		LS9999	

◆ **Elenco errori per ciascuna funzione**

Nome funzione Editor		Stato errore	Causa
_CF_dir ( )	LS9132	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella specificata con una stringa fissa)
		0012h	Errore nome file (nome percorso)
		0018h	Area LS entro l'intervallo di errore
		0020h	Nessuna CF-card
		0021h	CF-card non valida
		0100h	Errore apertura directory

Continua

<b>Nome funzione Editor</b>		<b>Stato errore</b>	<b>Causa</b>
_CF_read ( )	LS9133	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella/del file specificati con una stringa fissa)
		0011h	L'area LS legge l'intervallo di errore
		0012h	Errore nome file (nome percorso)
		0018h	Area LS entro l'intervallo di errore
		0020h	Nessuna CF-card
		0021h	CF-card non valida
		0101h	Errore ricerca file (Errore di offset)
		0102h	Numero di errori di estrapolazione byte
		0110h	Errore (aperto) di creazione file
_CF_write ( )	LS9134	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella/del file specificati con una stringa fissa)
		0011h	L'area LS legge l'intervallo di errore
		0012h	Errore nome file (nome percorso)
		0020h	Nessuna CF-card
		0021h	CF-card non valida
		0101h	Errore ricerca file (Errore di offset)
		0104h	Errore di creazione cartella
		0108h	Errore di modalità scrittura
		0110h	Errore (aperto) di creazione file
0111h	Errore di scrittura file (ad esempio, spazio insufficiente nella CF-card)		

Continua

<b>Nome funzione Editor</b>		<b>Stato errore</b>	<b>Causa</b>
_CF_delete ( )	LS9135	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella/del file specificati con una stringa fissa)
		0011h	L'area LS legge l'intervallo di errore
		0012h	Errore nome file (nome percorso)
		0020h	Nessuna CF-card
		0021h	CF-card non valida
		0112h	Errore di eliminazione file (ad esempio, il file specificato non esiste. Il file specificato è di sola lettura.)
_CF_rename ( )	LS9136	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella/del file specificati con una stringa fissa)
		0011h	L'area LS legge l'intervallo di errore
		0012h	Errore nome file (nome percorso)
		0020h	Nessuna CF-card
		0021h	CF-card non valida
		0114h	Errore di rinomina file (ad esempio, il file specificato non esiste. Il nome del file esiste già.)
_CF_read_csv ( )	LS9137	0001h	Errore parametro
		0002h	Errore CF-card (Nessuna CF-card, Errore di apertura file, Errore di lettura file)
		0003h	Errore di scrittura

Continua

<b>Nome funzione Editor</b>		<b>Stato errore</b>	<b>Causa</b>
__USB_dir ( )	LS9138	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella specificata con una stringa fissa)
		0012h	Errore nome file (nome percorso)
		0018h	Area LS entro l'intervallo di errore
		0020h	Nessuna memoria USB
		0021h	Memoria USB non valida
		0100h	Errore apertura directory
__USB_read ( )	LS9139	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella/del file specificati con una stringa fissa)
		0011h	L'area LS legge l'intervallo di errore
		0012h	Errore nome file (nome percorso)
		0018h	Area LS entro l'intervallo di errore
		0020h	Nessuna memoria USB
		0021h	Memoria USB non valida
		0101h	Errore ricerca file (Errore di offset)
		0102h	Numero di errori di estrapolazione byte
__USB_write ( )	LS9140	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella/del file specificati con una stringa fissa)
		0011h	L'area LS legge l'intervallo di errore
		0012h	Errore nome file (nome percorso)
		0020h	Nessuna memoria USB
		0021h	Memoria USB non valida
		0101h	Errore ricerca file (Errore di offset)
		0104h	Errore di creazione cartella
		0108h	Errore di modalità scrittura
		0110h	Errore (aperto) di creazione file
		0111h	Errore di scrittura file (ad esempio, spazio insufficiente nella memoria USB)

Continua

Nome funzione Editor		Stato errore	Causa
__USB_delete ( )	LS9141	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella/del file specificati con una stringa fissa)
		0011h	L'area LS legge l'intervallo di errore
		0012h	Errore nome file (nome percorso)
		0020h	Nessuna memoria USB
		0021h	Memoria USB non valida
		0112h	Errore di eliminazione file (ad esempio, il file specificato non esiste. Il file specificato è di sola lettura.)
__USB_rename ( )	LS9142	0010h	Dati D-Script non validi (Errore nel recuperare il nome della cartella/del file specificati con una stringa fissa)
		0011h	L'area LS legge l'intervallo di errore
		0012h	Errore nome file (nome percorso)
		0020h	Nessuna memoria USB
		0021h	Memoria USB non valida
		0114h	Errore di rinomina file (ad esempio, il file specificato non esiste. Il nome del file esiste già.)
__USB_read_csv ( )	LS9143	0001h	Errore parametro
		0002h	Errore memoria USB (Memoria USB assente, Errore apertura file, Errore lettura file)
		0003h	Errore di scrittura

## ◆ Modalità di memorizzazione dati

Quando i dati vengono letti/scritti dagli/agli indirizzi del dispositivo al momento di eseguire la funzione Lettura file/Scrittura file, l'ordine di memorizzazione dei dati scritti (estrapolati) può essere specificato.

L'impostazione della modalità di memorizzazione dati in LS9130 può cambiare l'ordine di memorizzazione. La modalità può essere selezionata tra quattro opzioni: 0, 1, 2 o 3.

### NOTA

- Usare quanto segue per fare riferimento a LS9130.
 

<code>_CF_write()</code>	operazione file CF: Scrivi file
<code>_CF_read()</code>	operazione file CF: Leggi file
<code>_CF_read_csv()</code>	operazione file CF: Leggi file CSV
<code>_USB_write()</code>	operazione file USB: Scrivi file
<code>_USB_read()</code>	operazione file USB: Leggi file
<code>_USB_read_csv()</code>	operazione file USB: Leggi file CSV
<code>IO_WRITE([p:PRN],...)</code>	operazione Stampante: Invia
- Quando si leggono gli indirizzi dei dispositivi oppure vi si scrive, sarà possibile usare le seguenti funzioni per interagire con la proprietà [Modalità Dati testo] nella pagina [Dispositivo/PLC] delle [Impostazioni di sistema], invece di usare la modalità Memoria LS9130.
 

<code>_CF_dir()</code>	operazione file CF: Elenco file output
<code>_USB_dir()</code>	operazione file USB: Elenco file output

### • Modalità 0

quando la funzione Lettura file viene utilizzata per scrivere una stringa "ABCDEFGH" nell'indirizzo di un dispositivo

`[w:[#INTERNAL]LS9130] = 0`

`_CF_read ("\"DATA\", \"DATA0001.BIN\", [w:[#INTERNAL]LS0100], 0, 7)`

- Quando la lunghezza dell'indirizzo del dispositivo è 16 bit

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

Scrivere "0" quando i dati da memorizzare consistono in un

- Quando la lunghezza dell'indirizzo del dispositivo è 32 bit

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

Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

### • Modalità 1

Ad esempio, quando la funzione Lettura file viene utilizzata per scrivere una stringa "ABCDEFGH" nell'indirizzo di un dispositivo

`[w:[#INTERNAL]LS9130] = 1`

`_CF_read ("\"DATA\", \"DATA0001.BIN\", [w:[#INTERNAL]LS0100], 0, 7)`

- Quando la lunghezza dell'indirizzo del dispositivo è 16 bit

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

Scrivere "0" quando i dati da memorizzare consistono in un

- Quando la lunghezza dell'indirizzo del dispositivo è 32 bit

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

Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

- Modalità 2

Ad esempio, quando la funzione Lettura file viene utilizzata per scrivere una stringa "ABCDEFGH" nell'indirizzo di un dispositivo

[w:[#INTERNAL]LS9130] = 2

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

- Quando la lunghezza dell'indirizzo del dispositivo è 16 bit

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

Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

- Quando la lunghezza dell'indirizzo del dispositivo è 32 bit

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

Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

- Modalità 3

Ad esempio, quando la funzione Lettura file viene utilizzata per scrivere una stringa "ABCDEFGH" nell'indirizzo di un dispositivo

[w:[#INTERNAL]LS9130] = 3

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

- Quando la lunghezza dell'indirizzo del dispositivo è 16 bit

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

Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

- Quando la lunghezza dell'indirizzo del dispositivo è 32 bit

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

Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

**IMPORTANTE**

- La modalità di memorizzazione dati non è identica alla modalità dati stringa nelle impostazioni del sistema. La relazione con la modalità dati stringa viene mostrata nella tabella seguente.

Ordine di memorizzazione del dispositivo dati	Ordine di memorizzazione doppia word LH/HL	Ordine di memorizzazione doppia word LH/HL Byte	Modalità di memorizzazione dei dati D-Script	Modalità dati del testo
Memorizza dai dati iniziali	Ordine HL	Ordine HL	0	1
	Ordine LH		1	2
	Ordine HL	Ordine LH	2	5
	Ordine LH		3	4
Memorizza dagli ultimi dati	Ordine HL	Ordine HL	–	3
	Ordine LH		–	7
	Ordine HL	Ordine LH	–	8
	Ordine LH		–	6

- Esiste un limite alla frequenza con cui i dati possono essere riscritti sulla CF-card. Pertanto, assicurarsi di eseguire regolarmente il backup di tutti i dati della CF-card su un altro supporto di memoria. Supponendo che 500 KB di dati in formato DOS siano sovrascritti, il limite è di 100.000 volte.
- Se si verifica un errore durante l'elaborazione della CF-card, l'errore verrà scritto nello stato di errori della CF-card/memoria USB [s:CF\_ERR\_STAT]/[s:USB\_ERR\_STAT].. Per altri dettagli, vedere " Stato errore CF-card/errore memoria USB" (pagina 21-110)
- I simboli e i caratteri riportati di seguito non possono essere utilizzati nei nomi delle cartelle o dei file. L'uso di questi simboli e caratteri nel nome di una cartella o di un file genererà un errore.

:	,	=	+	/	"	[
]		<	>	(spazio)	?	

- Per specificare una cartella principale (directory), specificare "" (stringa vuota) come nome della cartella.

## ■ Scrivi file

Elemento	Descrizione
Riepilogo	Si può selezionare una qualsiasi tra le tre modalità: "Nuovo/a", "Aggiungi" o "Sovrascrivi". Vedere la sezione "Modalità di memorizzazione dati" di seguito per ulteriori dettagli sull'ordine di memorizzazione dati.
Format	<p data-bbox="348 355 1188 417">_CF_write/_USB_write (nomi cartelle, nomi file, lettura dagli indirizzi, offset, numero di byte, modalità)</p> <div data-bbox="440 440 1094 842" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p data-bbox="348 871 477 896">Parametro 1</p> <p data-bbox="371 904 1177 966">Nome cartella: Stringa fissa (Lunghezza massima: 32 caratteri a byte singolo)</p> <p data-bbox="348 973 477 998">Parametro 2</p> <p data-bbox="371 1006 1204 1103">Nome file: Stringa fissa, Dispositivo interno, (Lunghezza massima: 32 caratteri a byte singolo), Dispositivo interno + Indirizzo temporaneo</p> <p data-bbox="348 1110 477 1136">Parametro 3</p> <p data-bbox="371 1143 1211 1205">Indirizzo Leggi-da: Indirizzo del dispositivo, Indirizzo del dispositivo + Indirizzo temporaneo</p> <p data-bbox="348 1213 477 1238">Parametro 4</p> <p data-bbox="371 1246 1232 1342">Offset: Valore numerico, Indirizzo dispositivo, Indirizzo temporaneo (Numero massimo specificato: 65535 per la lunghezza a 16 bit, 4.294.967.295 per la lunghezza a 32 bit)</p> <p data-bbox="348 1350 477 1375">Parametro 5</p> <p data-bbox="371 1383 1184 1445">Numero di byte: Valore numerico, Indirizzo del dispositivo, Indirizzo temporaneo (Lunghezza massima: 1280)</p> <p data-bbox="348 1452 477 1477">Parametro 6</p> <p data-bbox="371 1485 1232 1547">Modalità Valore numerico, Indirizzo dispositivo, Indirizzo temporaneo (Valori disponibili: 0, 1, 2)</p>

**Descrizione generale formato memoria**

Modalità	Nome	Descrizione
0	Nuovo	Creare un nuovo file. Se esiste un file con lo stesso nome, sarà eliminato.
1	Aggiungi	Aggiungere i dati a un file specifico. Se il file specifico non esiste, verrà creato un nuovo file.
2	Sovrascrivi	Sovrascrivere parte del file. Se l'offset specificato è più grande delle dimensioni del file, l'area in eccedenza sarà piena di 0 e i dati saranno scritti dopo l'area. Se l'offset viene specificato alla fine dei dati del file, l'operazione sarà equivalente all'aggiunta dei dati al file. Se il file non esiste, si verificherà un errore. Per altre informazioni su questa funzione, fare riferimento a " Stato errore CF-card/errore memoria USB" (pagina 21-110).

**Espressione esempio:**

```
[w:[#INTERNAL]LS0200] = 0      //Offset ("0" quando la modalità è "Nuovo/a")
[w:[#INTERNAL]LS0202] = 100   // Numero di byte (100 byte)
[w:[#INTERNAL]LS0204] = 0     // Modalità (Nuovo/a)
_CF_write ("\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100],
[w:[#INTERNAL]LS0200],
[w:[#INTERNAL]LS0202], [w:[#INTERNAL]:LS0204])
```

Nell'esempio precedente, 100 byte di dati sono letti da LS0100 e memorizzati nella cartella \DATA come DATA0001.BIN. Si può definire indirettamente il conteggio dei byte e la modalità definendo l'offset, il conteggio dei byte e la modalità con dispositivi interni.

**IMPORTANTE**

- L'impostazione offset è efficace solo nella modalità "Sovrascrivi".  
L'impostazione offset è disattivata nelle modalità "Nuovo/a" e "Aggiungi".  
Impostare il valore di offset su "0" in modalità diverse da "Sovrascrivi".
  - Se viene specificata la modalità "Nuovo/a" ed esiste un file con lo stesso nome, il file sarà sovrascritto.
  - Quando l'area LS viene specificata per "Nome file", "Indirizzo Leggi-da" non viene contata come indirizzo D-Script.
  - Quando un dispositivo PLC viene specificato per "Indirizzo Leggi-da", i dati vengono letti dal dispositivo PLC solo una volta quando la funzione viene eseguita. Se si verifica un errore durante la lettura dei dati, risulterà un errore di lettura della CF-card/memoria USB: [s:CF\_ERR\_STAT] o [s:<USB\_ERR\_STAT]. L'errore viene cancellato al completamento della lettura dei dati.
  - I dati saranno divisi in parti e letti dall'origine, benché questo dipenda dal numero di byte da leggere. Pertanto, anche se si verifica un errore di comunicazione durante la lettura dei dati, questi potrebbero essere stati in parte scritti nel file specificato.
  - Per specificare un percorso intero per un nome di file, specificare "\*" (asterisco) come nome della cartella.  
Ad esempio, `_CF_read ("*", "\DATA\DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 10)`
-

## Espressione esempio formato di memorizzazione

### ◆ Quando si specifica la modalità "Nuovo/a"

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

Nell'esempio sopra, 100 byte di dati vengono letti da LS0100 e il file DATA0001.BIN viene creato daccapo nella cartella \\DATA..

#### IMPORTANTE

- Per quanto riguarda il nome del file, si potrà utilizzare solo il formato 8.3 (un massimo di 12 caratteri, di cui 8 per il nome del file e 3 per l'estensione). Non si può utilizzare un nome di file più lungo di questo formato.

### ◆ Quando si specifica la modalità "Aggiungi"

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

Se il file specificato (DATA0001.BIN nell'esempio) già esiste e l'istruzione sopra viene eseguita, verranno letti 100 byte di dati da LS0100 e aree seguenti, e saranno aggiunti al file DATA0001.BIN nella cartella \\DATA.

### ◆ Quando si specifica la modalità "Sovrascrivi" (1)

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

Se il file specificato (DATA0001.BIN nell'esempio) già esiste e l'istruzione sopra viene eseguita, verranno letti 10 byte di dati da LS0100 e aree seguenti e sovrascritti 10 byte di dati memorizzati nel bit 16 e quelli seguenti dopo l'offset nel file DATA0001.BIN nella cartella \\DATA.

### ◆ Quando si specifica la modalità "Sovrascrivi" (2)

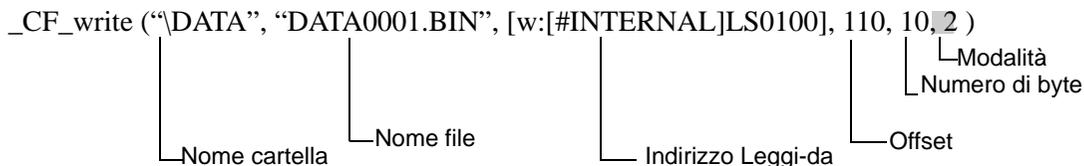
(Il file da sovrascrivere è inferiore alla somma del valore di offset e del numero di byte sommati).

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

Il file specificato (nell'esempio, DATA0001.BIN) esiste già e le sue dimensioni corrispondono a 100 byte. Quando l'offset viene impostato su 96 byte e il numero di byte è impostato su 10 byte per l'operazione di sovrascrittura, vengono letti 10 byte di dati memorizzati in LS0100 e aree seguenti. Quindi, i primi 4 byte di dati estrapolati sovrascrivono i 4 byte di dati memorizzati nel 97mo byte e in quelli seguenti nel file, mentre i rimanenti 6 byte di dati saranno aggiunti alla fine dei dati del file. Il file risultante contiene 106 byte di dati.

◆ **Quando si specifica la modalità "Sovrascrivi" (3)**

(Il file da sovrascrivere è inferiore al valore di offset.)



Il file specificato (nell'esempio, DATA0001.BIN) esiste già e le sue dimensioni corrispondono a 100 byte. Quando l'offset è impostato su 110 byte e il numero di byte è impostato su 10 byte per l'operazione di sovrascrittura, l'area tra il 101mo byte si riempie di 0 e i 10 byte di data letti da LS0100 e aree seguenti sono scritti nel 111mo e byte seguenti. Il file risultante contiene 120 byte di dati.

**IMPORTANTE**

- Il massimo numero consentito di caratteri per il primo parametro (Nome cartella) e il secondo parametro (Nome file) è pari a 32 caratteri a byte singolo.
- Il dispositivo interno può essere specificato per il secondo parametro, (Nome file). Specificare il dispositivo interno consente l'indirizzamento indiretto di un nome file. Inoltre, un numero di caratteri fino a 32 a byte singolo può essere utilizzato per specificare il nome di un file.  
 Ad esempio, `_CF_write ("DATA", [w:[#INTERNAL]LS0100], [w:[#INTERNAL]LS0200], 0, 100, 0)`  
 Memorizzare il nome di un file in LS0100 consente l'indirizzamento indiretto per un nome di un file. In quest'esempio, un nome di file viene memorizzato in LS0100 mediante LS0106 come segue.

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

La fine del nome del file deve essere un carattere NULL. Il dispositivo di visualizzazione riconosce i dati prima del carattere NULL come nome del file.

Nell'esempio riportato sopra, 100 byte di dati vengono letti da LS0200 e viene creato il nuovo file "\DATA\DATA0001.BIN" per memorizzare i dati.

- Per quanto riguarda il nome del file, si potrà utilizzare solo il formato 8.3 (un massimo di 12 caratteri, di cui 8 per il nome del file e 3 per l'estensione). Non si possono usare nome di file superiori a questa lunghezza.

## ■ Cambia nome file

Elemento	Descrizione
Riepilogo	Modifica il nome del file. Il Parametro 1 designa la cartella con i dati della CF-card. Il Parametro 2 designa il nome originale del file. Il Parametro 3 designa il nuovo nome del file.
Format	<p>_CF_rename/_USB_rename (nomi cartelle, nomi file, nomi file cambiati)  Il nome del file può, inoltre, essere designato indirettamente con l'indirizzo LS.</p> <div data-bbox="454 479 1108 788" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Parametro 1  Nome cartella: Testo fisso</p> <p>Parametro 2  Nome file: Testo fisso, Dispositivo interno, Dispositivo interno + Indirizzo temporaneo</p> <p>Parametro 3  Nome file: Testo fisso, Dispositivo interno, Dispositivo interno + Indirizzo temporaneo</p>

**Espressione esempio:**

```
_CF_rename ("\\DATA", "DATA0001.BIN", "DATA1234.BIN")
```

Nell'esempio riportato sopra il nome del file cambia da “\\DATA\\DATA0001.BIN” a “\\DATA\\DATA1234.BIN”.

**IMPORTANTE**

- Per quanto riguarda il nome del file, si potrà utilizzare solo il formato 8.3 (un massimo di 12 caratteri, di cui 8 per il nome del file e 3 per l'estensione). Non si possono usare nomi di file superiori a questa lunghezza.
- Il massimo numero consentito di caratteri per il primo parametro (Nome cartella) e il secondo parametro (Nome file) è pari a 32 caratteri a byte singolo.
- Il dispositivo interno può essere specificato per il secondo e terzo parametro (Nomi file). Specificare il dispositivo interno consente l'indirizzamento indiretto di un nome file. Inoltre, un numero di caratteri fino a 32 a byte singolo può essere utilizzato per specificare il nome di un file.

Esempio

```
_CF_rename ("\\DATA", [w:[#INTERNAL]LS0100],  
[w:[#INTERNAL]LS0200])
```

Memorizzare il nome di un file in LS0100 e LS0200 consente l'indirizzamento indiretto per un nome di un file.

- Memorizzare i nomi dei file in LS0100 mediante LS0106 come segue.

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

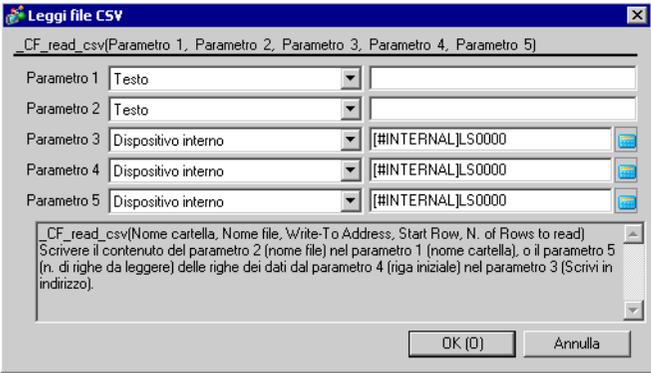
La fine del nome del file deve essere un carattere NULL. L'unità GP riconosce i dati prima del carattere

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

Nell'istruzione precedente, il file “\\DATA\\DATA0001.BIN” viene rinominato in “\\DATA\\DATA1234.BIN”.

- Quando l'area LS viene specificata per “Nome file”, non viene contata come indirizzo D-Script.
- Per specificare una cartella principale (directory), specificare “ ” (stringa vuota) come nome della cartella.
- Per specificare un percorso intero per un nome di file, specificare “\*” (asterisco) come nome della cartella.

## ■ Leggi file CSV

Elemento	Descrizione
Riepilogo	Legge i dati in unità celle da un file CSV (costruiti da un'immagine della cella delimitata da “,”), e li scrive in un indirizzo word.
Format	<p>_CF_read_csv/_USB_read_csv (nomi cartelle, nomi file, indirizzi salva-in, linea iniziale, numero di linee lette)</p>  <p>Parametro 1: Testo (fino a 32 caratteri a bit singolo)  Parametro 2: Testo (fino a 32 caratteri a byte singolo), Dispositivo interno, Dispositivo interno + Indirizzo temporaneo  Parametro 3: Dispositivo interno, Dispositivo interno designato con offset  Parametro 4: Valore numerico (da 0 a 65535), Dispositivo interno, Variabile temporanea  Parametro 5: Valore numerico (da 0 a 65535), Dispositivo interno, Variabile temporanea</p>

### Espressione esempio:

\_CF\_read\_csv (“\CSV”, “SAMPLE.CSV”, [w:[#INTERNAL]LS1000], 1, 2)

(Quando si leggono due linee di dati, iniziando dalla prima linea del file [CSV\SAMPLE.CSV] nella CF-card della memoria utilizzando la funzione “\_CF\_read\_csv ()” .)

SAMPLE.CSV

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

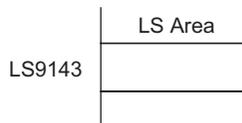
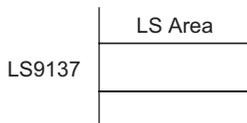
Legge due linee di dati, iniziando dalla prima linea del file CSV. Quando il primo carattere è un valore numerico (da “0” a “9” o “-”), i dati sono memorizzati come valore numerico. Quando il primo carattere è [“”], i dati vengono trattati come un carattere e “00h” viene memorizzato alla fine della stringa di testo. Ad esempio, quando si memorizza “DAT01-01”, la dimensione dei dati è di 8 caratteri, un numero pari, e viene utilizzato un totale di cinque word: quattro word sono utilizzate per memorizzare la stringa di testo, e una word è utilizzata per memorizzare “00h” alla fine. Ad esempio quando si memorizza “DAT01-2”, le dimensioni dei dati equivalgono a 7 caratteri, numero dispari, mentre per memorizzare il testo si usa un totale di 4 word, con “00h” memorizzato alla fine.

		16 bit	
	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

Quando la modalità di memorizzazione dati è 0

#### NOTA

- Quando il primo carattere della cella è un valore numerico (da “0” a “9” o “-”), converte il valore dei dati numerici e quindi scrive i dati nel dispositivo LS. L'intervallo consentito va da -32768 a 32767.
- Quando il primo carattere nella cella è [“”], scrive l'intervallo con [“”] nel dispositivo LS come dati stringhe di testo. Quando le dimensioni dei dati stringhe di testo consistono in un numero dispari di byte, “0x00” viene aggiunto alla fine. Quando le dimensioni dei dati del testo consistono in un numero pari di byte, “0x0000” viene scritto nell'indirizzo successivo rispetto all'ultimo indirizzo. Nella cella si possono inserire fino a 32 caratteri a byte singolo.
- Quando un file CSV presenta due o più linee di dati, il numero desiderato di linee può essere letto iniziando dalla linea specificata. In una linea possono essere inseriti fino a 200 caratteri a byte singolo, mentre il numero massimo di linee da inserire in un file CSV è 65535.
- Quando si verifica un errore, lo stato di errore viene scritto in LS9137 (LS9143 per la memoria USB).
- Quando si scrivono dati di testo dei file CSV nel dispositivo LS, l'ordine di memorizzazione dei dati dipende dalla modalità di memorizzazione dei dati.

**Stato errore**

Nome funzione Editor	Area LS	Stato errore	Causa
_CF_read_csv ()/ _USB_read_csv ()	LS9137/ LS9143	0000h	Fase di completamento riuscita
		0001h	Errore parametro
		0002h	Errore CF-card/Memoria USB Nessuna CF-card o memoria USB/Errore apertura file/Errore lettura file
		0003h	Errore lettura/scrittura

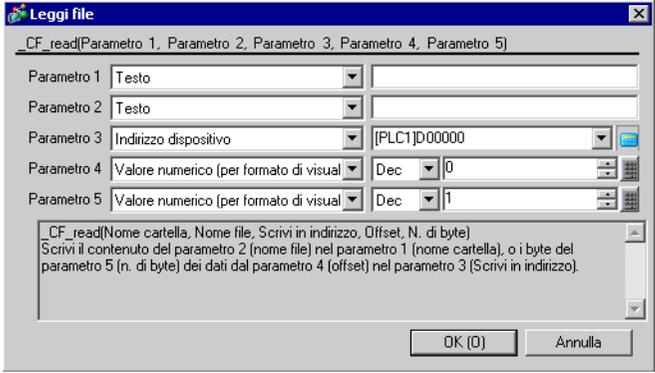
**IMPORTANTE**

- Quando " \* " viene specificato per il nome della cartella, l'intero percorso può essere designato per il nome del file.
- Per quanto riguarda il nome del file, si potrà utilizzare solo il formato 8.3 (un massimo di 12 caratteri, di cui 8 per il nome del file e 3 per l'estensione). Non si può utilizzare un nome di file più lungo di questo formato.
- L'area effettiva per la memorizzazione di dati importati da un file CSV nel dispositivo LS si limita all'area designata per l'utente (da LS20 a LS2031 e da LS2096 a LS8191).
- Il periodo di elaborazione richiesto per l'importazione di dati è proporzionale al volume dei dati del file CSV da estrapolare. Le parti non saranno aggiornate fino al completamento dell'elaborazione. (Vengono impiegati circa 10 secondi per leggere i dati dalla prima alla 100ma linea di un file CSV contenente 100 linee, con 40 caratteri per linea.)
- A differenza della funzione "\_CF()/\_USB\_read ()", lo stato non sarà salvato in [s:CF\_ERR\_STAT]/[s:USB\_ERR\_STAT] immediatamente dopo l'esecuzione della funzione. (In alcuni casi, potrebbero essere inseriti valori non definiti.)
- Assicurarsi di inserire [" ] all'inizio e alla fine delle stringhe di testo che iniziano con un numerale.

Ad esempio:

```
[ 123, 2-D4EA ] [ 123, "2-D4EA" ]
      X           O
```

## ■ Leggi file

Elemento	Descrizione
Riepilogo	Legge il numero specificato di byte di dati nel file dopo l'offset specificato, e lo scrive nell'indirizzo di destinazione. Vedere la sezione “Modalità di memorizzazione dati” di seguito per ulteriori dettagli sull'ordine di memorizzazione dati.
Format	<p>_CF_write/_USB_write (nomi cartelle, nomi file, salvataggio in indirizzi, offset, numero di byte)</p>  <p>Parametro 1 Nome cartella: Stringa fissa (Lunghezza massima: 32 caratteri a byte singolo)</p> <p>Parametro 2 Nome file: Stringa fissa, Dispositivo interno, Dispositivo interno + Indirizzo temporaneo (Lunghezza massima: 32 caratteri a byte singolo)</p> <p>Parametro 3 Indirizzo Scrivi-in: Indirizzo del dispositivo + Indirizzo temporaneo</p> <p>Parametro 4 Offset: Valore numerico, Indirizzo dispositivo, Indirizzo temporaneo (Numero massimo specificato: 65535 per la lunghezza a 16 bit, 4.294.967.295 per la lunghezza a 32 bit)</p> <p>Parametro 5 Numero di byte: Valore numerico, Indirizzo del dispositivo, Indirizzo temporaneo (Lunghezza massima: 1280)</p>

### Espressione esempio:

Per leggere 16 byte di dati nel file specificato quando l'offset è 16:

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

Nell'esempio riportato sopra, 16 byte di dati dai 17 byte in poi nel file

“\DATA\DATA0001.BIN” vengono letti in LS0100 e dalle aree seguenti.

**IMPORTANTE**

- Per quanto riguarda il nome del file, si potrà utilizzare solo il formato 8.3 (un massimo di 12 caratteri, di cui 8 per il nome del file e 3 per l'estensione). Non si possono usare nomi di file superiori a questa lunghezza.
- Il massimo numero consentito di caratteri per il primo parametro (Nome cartella) e il secondo parametro (Nome file) è pari a 32 caratteri a byte singolo.
- Il dispositivo interno può essere specificato per il secondo parametro, (Nome file). Specificare il dispositivo interno consente l'indirizzamento indiretto di un nome file. Inoltre, un numero di caratteri fino a 32 a byte singolo può essere utilizzato per specificare il nome di un file.

**Esempio**

Per leggere 10 byte di dati memorizzati in un file quando il file viene specificato in LS0100 e oltre, e l'offset è 0:

```
_CF_read ("DATA", [w:LS0100], [w:LS0200], 0, 10)
```

Memorizzare il nome di un file in LS0100 consente l'indirizzamento indiretto per un nome di un file. In quest'esempio, un nome di file viene memorizzato in LS0100 mediante LS0106 come segue.

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

← La fine del nome del file deve essere un carattere NULL. Il dispositivo di visualizzazione riconosce i dati prima del carattere NULL come nome del file.

Nell'esempio riportato sopra, 10 byte di dati all'inizio del file "DATA\DATA0001.BIN" vengono letti in LS0200 e dalle aree seguenti.

- Il numero di byte che sono stati letti correttamente viene scritto in byte estrapolati della nella CF-card/memoria USB [s:CF\_READ\_NUM]/[s:USB\_READ\_NUM]. Per altri dettagli, vedere "21.11.5 Operazione file CF/ Operazione file USB Stato errore CF-card/errore memoria USB" (pagina 21-110)
- Il dispositivo interno designato in "Nome file" e l'"Indirizzo Scrivi -a" non vengono contati come indirizzi D-Script.
- Quando si specifica un dispositivo PLC per l'Indirizzo Scrivi -a, si impiega più tempo a scrivere i dati nel dispositivo PLC, poiché il numero di word (byte) aumenta. Potrebbero trascorrere parecchi secondi per scrivere i dati, in base al numero di word.
- Se i dati estrapolati dal file superano l'intervallo designato del dispositivo PLC, si verificherà un errore di comunicazione. In questo caso, si dovrà spegnere il dispositivo PLC e poi riaccenderlo per azzerare l'errore.

Continua

**IMPORTANTE**

- Quando un dispositivo PLC viene specificato come destinazione, i valori non sono scritti immediatamente, a causa del tempo di trasmissione dall'unità GP al dispositivo PLC.

Esempio

Nello script riportato di seguito, l'istruzione (1) legge 10 byte di dati dal file e li scrive in [w:D0100]. In ogni caso, i dati non sono ancora stati scritti in [w:[PLC1]D0100] all'esecuzione dell'istruzione (2) a causa del tempo di trasmissione.

```
_CF_read ("\\DATA", "DATA0001.BIN", [w:[PLC1]D0100], 0, 10) ..... (1)  
[w:[PLC1]D0200] = [w:[PLC1]D0100] + 1 .....(2)
```

In questo caso, memorizzare i dati una volta sola nell'area LS ed eseguire la seconda istruzione come segue.

```
_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
```

---

## ■ Elenco file output

Elemento	Descrizione
Riepilogo	<p>L'elenco di file esistenti nella cartella specificata viene scritto nel dispositivo interno. Il Parametro 1 designa la cartella con i dati della CF-card. Il Parametro 4 designa l'offset utilizzato per selezionare un file/alcuni file in quella cartella. Il Parametro 3 designa il numero di file selezionati nella cartella citata. Il Parametro 2 specifica l'area LS in cui i file saranno scritti. Quando l'offset viene specificato come "0", l'elenco inizia dal primo file (iniziale).</p>
Format	<p>_CF_dir/_USB_dir (nomi cartelle, indirizzi salva-in, numero di file, offset)</p> <div data-bbox="460 577 1112 919" data-label="Image"> </div> <p>Parametro 1 Nome cartella: Testo fisso (Lunghezza massima: 32 caratteri a byte singolo)</p> <p>Parametro 2 Indirizzo Scrivi-in: Dispositivo interno, Dispositivo interno designato con offset</p> <p>Parametro 3 Numero di file: Valore numerico, Indirizzo del dispositivo, Indirizzo temporaneo (Lunghezza massima: 32)</p> <p>Parametro 4 Valore numerico, Indirizzo dispositivo, Indirizzo temporaneo</p>

**Espressione esempio:**

Per emettere un elenco file contenente due file quando l'offset è 1 (secondo file):

```
_CF_dir ("\\DATA\\*.*", [w:[#INTERNAL]LS0100], 2, 1)
```

Quando l'istruzione sopra indicata viene eseguita mentre i seguenti file esistono nella cartella DATA, i nomi dei file "DATA0001.BIN" e "DATA02.BIN" sono scritti in LS0100 e aree successive.

Contenuto della cartella		Contenuto dell'area LS			
		16 bit			
/DATA	DATA0000.BIN	LS0100	'D' 'A'	} 7 word usate.	
	DATA0001.BIN	LS0101	'T' 'A'		
	DATA02.BIN	LS0102	'0' '0'		
	DATA003.BIN	LS0103	'0' '1'		
	DATA0004.BIN	LS0104	'.' 'B'		
		LS0105	'I' 'N'		} 7 word usate.
		LS0106	'\0' '\0'		
		LS0107	'D' 'A'		
		LS0108	'T' 'A'		
		LS0109	'0' '2'		
		LS0110	'.' 'B'		
		LS0111	'I' 'N'		
		LS0112	'\0' '\0'		
		LS0113	'\0' '\0'		

**IMPORTANTE**

- Quando l'offset viene specificato come "0", l'elenco inizia dal primo file (iniziale).
- Per quanto riguarda il nome del file, si potrà utilizzare solo il formato 8.3 (un massimo di 12 caratteri, di cui 8 per il nome del file e 3 per l'estensione). Non si possono usare nomi di file superiori a questa lunghezza.
- Se la cartella specificata non contiene abbastanza file, come specificato, la rimanente area LS si riempirà di caratteri NULL ('\0').
- Se un nome di file ha meno di 12 caratteri, le posizioni vuote si riempiranno di caratteri NULL ('\0').
- Per specificare un nome di cartella, quale "\\DATA\\\*.\*", accertarsi di aggiungere "\*.\*". Gli asterischi \*.\* indicano di elencare tutti i file.
- Il numero di file effettivamente elencati viene scritto nei file elencati nella CF-card /memoria USB [s:CF\_FILELIST\_NUM]/[s:USB\_FILELIST\_NUM]. Per informazioni dettagliate, fare riferimento a "Stato errore CF-card/errore memoria USB" (pagina 21-110)
- Gli indirizzi LS Scrivi-in non vengono contati come indirizzi D-Script.
- I nomi dei file non sono ordinati quando vengono scritti nell'area LS. Sono scritti in ordine di creazione (l'ordine di inserimento FAT).
- L'elenco può essere creato specificando un'estensione file. Per elencare file con una determinata estensione, usare un formato come "\\DATA\\\*.BIN". In ogni caso, non si può usare "\*" nel nome di un file.

## ■ Elimina file

Elemento	Descrizione
Riepilogo	Elimina il file specificato dalla CF-card. Il Parametro 1 designa la cartella con i dati della CF-card. Il Parametro 2 designa il numero di file da eliminare.
Format	<p>_CF_delete/_USB_delete (nomi cartelle, nomi file)          Il nome del file può, inoltre, essere designato indirettamente con l'indirizzo LS.</p> <div data-bbox="495 484 1149 794" style="border: 1px solid gray; padding: 5px; margin: 10px auto; width: fit-content;"> </div> <p>Parametro 1          Nome cartella: Testo fisso</p> <p>Parametro 2          Nome file: Testo fisso, Dispositivo interno, Dispositivo interno + Indirizzo temporaneo</p>

### Espressione esempio:

\_CF\_delete ("DATA", "DATA0001.BIN")

L'esempio sopra elimina il file "\DATA\DATA0001.BIN".

**IMPORTANTE**

- Per quanto riguarda il nome del file, si potrà utilizzare solo il formato 8.3 (un massimo di 12 caratteri, di cui 8 per il nome del file e 3 per l'estensione). Non si possono usare nomi di file superiori a questa lunghezza.
- Il massimo numero consentito di caratteri per il primo parametro (Nome cartella) e il secondo parametro (Nome file) è pari a 32 caratteri a byte singolo.
- Il dispositivo interno può essere specificato per il secondo parametro, (Nome file). Specificare il dispositivo interno consente l'indirizzamento indiretto di un nome file. Inoltre, un numero di caratteri fino a 32 a byte singolo può essere utilizzato per specificare il nome di un file.

In quest'esempio, un nome di file viene memorizzato in LS0100 mediante LS0106 come segue.

16 bit

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

La fine del nome del file deve essere un carattere NULL. Il dispositivo di visualizzazione riconosce i dati prima del carattere NULL come nome del file.

L'esempio sopra elimina il file "\DATA\DATA0001.BIN".

- Per specificare una cartella principale (directory), specificare "" (stringa vuota) come nome della cartella.
- Quando l'area LS viene specificata per "Nome file", "Indirizzo Scrivi-in" non viene contata come indirizzo D-Script.
- Per specificare un percorso intero per un nome di file, specificare "\*" (asterisco) come nome della cartella.

## 21.11.6 Operazione stampante

Operazione stampante	Riepilogo funzioni
	<p>Impostazioni etichette   " ■ Impostazioni etichette" (pagina 21-136)  Designata dalle variabili Controllo e Stato.</p> <hr/> <p>Invia   " ■ Invia" (pagina 21-138)  Emette il numero designato di byte nella porta COM.</p>

**IMPORTANTE**

- COM1 o USB/PIO (USB-PIO) sono porte che possono essere utilizzate come Funzione operazioni stampante.

### ■ Impostazioni etichette

#### Controllo

La variabile Controllo (PRN\_CNTRL) viene utilizzata per azzerare il buffer Invia, il buffer Ricevi e lo stato di errore. Questa variabile è di sola scrittura.

- Riepilogo (PRN\_CTRL) Controllo

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<input type="checkbox"/>															

Bit	Contenuto
15	
14	
13	
12	
11	
10	
9	Riservato
8	
7	
6	
5	
4	
3	
2	1: Azzerare errore
1	Riservato
0	1: Azzerare buffer invio

**IMPORTANTE**

- Quando si seleziona una word e due o più bit sono impostati simultaneamente, l'elaborazione verrà eseguita nell'ordine indicato di seguito:

Azzerare errore



Azzerare buffer di invio

- Non usare bit riservati. Imposta solo i bit necessari.

## Stato

La variabile Stato (PRN\_STAT) viene utilizzata per controllare la presenza/l'assenza di dati del buffer di invio e per ottenere lo stato di errore. Questa variabile Stato è di sola scrittura.

- Contenuto della variabile Stato (PRN\_STAT)

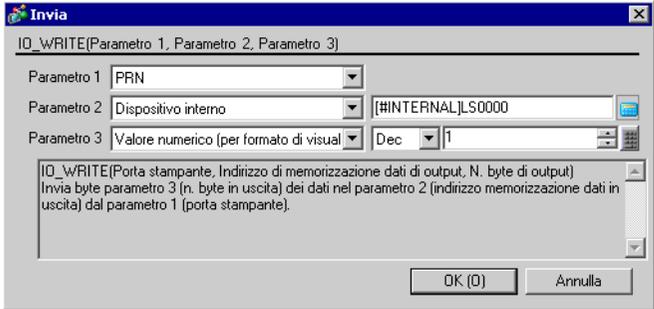
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Contenuto
15	Riservato
14	Lo stato del segnale ERROR I/F della stampante Errore stampante (Ingresso): 0: Errore 1: Normale
13	Lo stato del segnale SLCT I/F della stampante Selezione (Ingresso): 0: Non in linea 1: Online
12	Lo stato del segnale PE I/F della stampante Carta esaurita (Input): 0: Normale Carta esaurita (Ingresso):
11	Riservato
10	
9	
8	
7	
6	
5	
4	
3	
2	
1	0: Normale 1: Invia errore
0	0: Data exists in Send buffer 1: Invia buffer è vuoto

### IMPORTANTE

- Se il buffer Invio è sovraccarico, si verificherà un errore. Quando si verifica questo errore, il bit di trasmissione si porterà su ON.
- Il buffer di invio consta di 8192 byte.
- I bit riservati potrebbero essere assegnati in futuro. Pertanto, assicurarsi di selezionare solo i bit necessari.

## ■ Invia

Elemento	Descrizione
Riepilogo	Emette il numero designato di byte nella porta COM. I dati sono emessi in ogni caso, indipendentemente dal tipo di stampante specificato.
Format	<p>IO_WRITE ([p:PRN]Indirizzo di memorizzazione dati emessi, Numero di byte emessi)</p>  <p>Parametro 1: [p:PRN]          Parametro 2: Dispositivo interno          Parametro 3: Valore intero, Indirizzo dispositivo, Indirizzo temporaneo</p>

### IMPORTANTE

- Il valore massimo da specificare per il Parametro 3 è 1024. Anche quando si specificano valori superiori a 1024, saranno emessi solo 1024 byte di dati dalla porta COM.

### Espressione esempio 1:

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

Nell'esempio sopra, 10 byte di dati memorizzati in LS1000 ed aree successive sono emessi dalla porta COM.

### Espressione esempio 2:

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

Nell'esempio sopra, i dati memorizzati in LS1000 ed aree successive sono emessi dalla porta COM. Il numero di byte è lo stesso di quello scritto in LS0800.

### Espressione esempio 3:

IO\_WRITE ([p:PRN], [w:[#INTERNAL]LS 1000], [t:0010])

Nell'esempio sopra, i dati memorizzati in LS1000 ed aree successive sono emessi dalla porta COM. Il numero di byte è lo stesso di quello scritto nell'indirizzo temporaneo [t:0010].

## Modalità di memorizzazione dati

Quando i dati vengono letti dagli indirizzi dei dispositivi su esecuzione della funzione Operazione porta COM, si può specificare l'ordine di memorizzazione dei dati estrapolati. L'impostazione della modalità di memorizzazione dati in LS9130 può cambiare l'ordine di memorizzazione.

La modalità può essere selezionata tra quattro opzioni: 0, 1, 2 o 3.

### ◆ Modalità 0

Ad esempio, quando la funzione Operazione porta COM viene usata per leggere la stringa "ABCDEFGH" nell'indirizzo di un dispositivo

$[w:[\#INTERNAL]LS9130] = 0$

`IO_WRITE ([p:PRN], [w:[\#INTERNAL]LS1000], 7)`

- Quando la lunghezza dell'indirizzo del dispositivo è 16 bit

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

← Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

- Quando la lunghezza dell'indirizzo del dispositivo è 32 bit

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

← Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

### ◆ Modalità 1

Ad esempio, quando la funzione Operazione porta COM viene usata per leggere la stringa "ABCDEFGH" nell'indirizzo di un dispositivo

$[w:[\#INTERNAL]LS9130] = 1$

`IO_WRITE ([p:PRN], [w:[\#INTERNAL]LS1000], 7)`

- Quando la lunghezza dell'indirizzo del dispositivo è 16 bit

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

← Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

- Quando la lunghezza dell'indirizzo del dispositivo è 32 bit

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

← Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

◆ **Modalità 2**

Ad esempio, quando la funzione Operazione porta COM viene usata per leggere la stringa “ABCDEFGH” nell’indirizzo di un dispositivo

[w:[#INTERNAL]LS9130] = 2

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

- Quando la lunghezza dell’indirizzo del dispositivo è 16 bit

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



Scrivere “0” quando i dati da memorizzare consistono in un numero dispari di byte.

- Quando la lunghezza dell’indirizzo del dispositivo è 32 bit

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



Scrivere “0” quando i dati da memorizzare consistono in un numero dispari di byte.

◆ **Modalità 3**

Ad esempio, quando la funzione Operazione porta COM viene usata per leggere la stringa “ABCDEFGH” nell’indirizzo di un dispositivo

[w:[#INTERNAL]LS9130] = 3

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

- Quando la lunghezza dell’indirizzo del dispositivo è 16 bit

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



Scrivere “0” quando i dati da memorizzare consistono in un numero dispari di byte.

- Quando la lunghezza dell'indirizzo del dispositivo è 32 bit

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

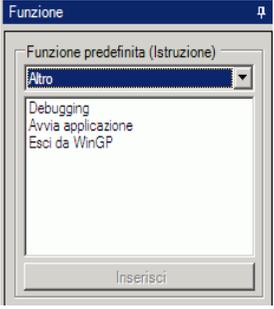
← Scrivere "0" quando i dati da memorizzare consistono in un numero dispari di byte.

**IMPORTANTE**

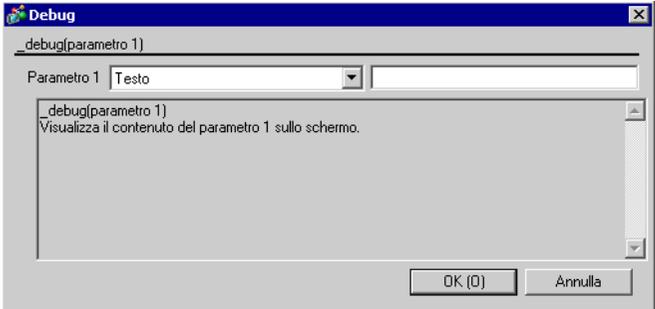
- La modalità di memorizzazione dati non è identica alla modalità dati stringa nelle impostazioni del sistema. La relazione con la modalità dati stringa viene mostrata nella tabella seguente.

Ordine di memorizzazione del dispositivo dati	Ordine di memorizzazione doppia word LH/HL	Ordine di memorizzazione doppia word LH/HL Byte	Modalità di memorizzazione dei dati D-Script	Modalità dati del testo
Memorizza dai dati iniziali	Ordine HL	Ordine HL	0	1
	Ordine LH		1	2
	Ordine HL	Ordine LH	2	5
	Ordine LH		3	4
Memorizza dagli ultimi dati	Ordine HL	Ordine HL	-	3
	Ordine LH		-	7
	Ordine HL	Ordine LH	-	8
	Ordine LH		-	6

## 21.11.7 Altri

Altro	Riepilogo funzioni
	<p><b>Funzione debugging</b>   " ■ Funzione debugging" (pagina 21-142)            Visualizza l'indirizzo o il testo designato sullo schermo per eseguirne il debugging.</p>
	<p><b>Attivazione dell'applicazione</b>   " ■ Attivazione dell'applicazione" (pagina 21-144)            Esegue l'intervallo specificato e avvia l'applicazione.</p>
	<p><b>WinGP, uscita</b>   " ■ Esci da WinGP" (pagina 21-146)            Uscire da WinGP.</p>

## ■ Funzione debugging

Elemento	Descrizione
Riepilogo	<p>Visualizza l'indirizzo o il testo designato sullo schermo per eseguirne il debugging.</p> <p>Dopo aver terminato il debugging e azzerato la casella [Attiva funzione debugging] dell'Editor script, nessuno degli script sarà eliminato. Solo lo schermo di debugging non apparirà.</p>
Format	<p><code>_debug (Parametro 1)</code></p>  <p>Parametro 1: Testo (fino a 32 caratteri a bit singolo, 16 caratteri a doppio byte)</p>

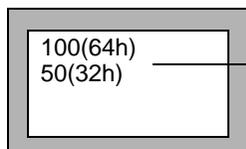
**Contenuto del Parametro 1**

Parametro 1	Format	Descrizione
Testo	_debug ("ABC")	Visualizza il testo interno " ". Il testo può essere lungo fino a 32 caratteri a byte singolo.
Indirizzo word o Indirizzo temporaneo	_debug (w:[PLC1]D1000)	Visualizza il valore dell'Indirizzo word o dell'Indirizzo temporaneo impostati.
Salto riga	_debug (_CRLF)	Sposta il cursore all'inizio della riga successiva.
Ritorno a capo	_debug (_CR)	Sposta il cursore all'inizio della stessa riga.

**◆ Espressione esempio 1:**

Lo script successivo visualizza il valore dell'Indirizzo word.

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



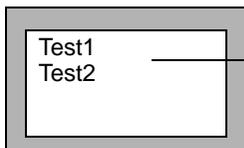
Il visualizzatore è nel formato seguente.  
\*\*\*\*\* (\*\*h)

Decimale Esadecimale

**◆ Espressione esempio 2:**

Lo script successivo mostra un ritorno a capo e il testo.

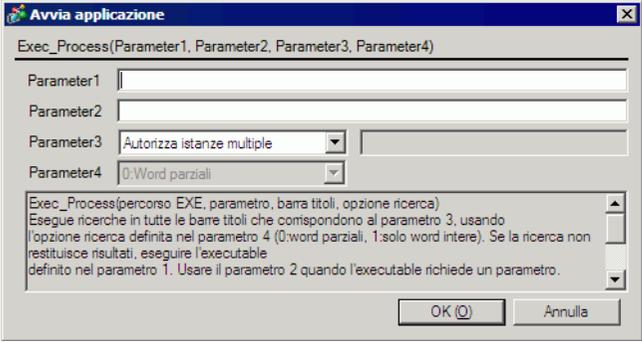
```
_debugging ("Test1")
_debug (_CRLF)
_debugging ("Test2")
```



Scorrere di una linea verso il basso e visualizzare "Test 2".

## ■ Attivazione dell'applicazione

Questa funzione è operativa solo sui modelli della Serie IPC.

Elemento	Descrizione
Riepilogo	<p>Esegue l'intervallo specificato e avvia l'applicazione. È possibile specificare impostazioni quali i parametri di avvio e il controllo nell'avvio multiplex.</p>
Format	<p>Processo Esec (Parametro 1, Parametro 2, Parametro 3, Parametro 4)</p>  <p>Parametro 1 Percorso EXE: Inserire il percorso assoluto del file eseguibile (.exe) per l'applicazione che si desidera avviare. Si possono inserire fino a 255 caratteri.</p> <p>Parametro 2 Parametro: Inserire l'argomento di avvio del file eseguibile. Si possono inserire fino a 255 caratteri.</p> <p>Parametro 3 Titolo finestra: Se non si desidera consentire istanze multiple, selezionare "Non consentire istanze multiple" e inserire il [Titolo finestra]. Si possono inserire fino a 63 caratteri. L'applicazione non può iniziare se viene trovata una finestra uguale a [Titolo finestra]. Sono consentite istanze multiple se si seleziona [Autorizza istanze multiple] o se il [Titolo finestra] non è stato specificato.</p> <p>Parametro 4 Trova solo titoli di finestra interi: Attivato solo quando si seleziona il Parametro3 - "Non consentire istanze multiple". Quando si seleziona "0: Word parziali", l'applicazione specificata non viene eseguita se viene trovata una finestra dal titolo simile a quello specificato in [Titolo finestra]. Quando si seleziona "1: Solo word intere", l'applicazione specificata non viene eseguita se viene trovata una finestra dal titolo esattamente uguale a quello specificato in [Titolo finestra].</p>

**NOTA**

- Il Parametro1 richiede l'inserimento di testo (percorso EXE). Si verifica un errore se non si inserisce del testo.
- Questa funzione è operativa solo sui modelli della Serie IPC.

**Metodo di input del Parametro 1 (percorso EXE)**

Il percorso EXE può essere inserito in tre modi diversi:

Nel seguente esempio è descritta l'esecuzione del file sample.exe in C:\Documents and Settings\utente\Impostazioni locali\Temp.

## 1. Specifica del percorso completo

Ad esempio, C:\Documents and Settings\user\Local Settings\Temp\sample.exe

## 2. Solo il nome del file EXE

Questa operazione è possibile se il file eseguibile si trova in una cartella che è stata specificata come percorso delle Impostazioni ambiente su un dispositivo della Serie IPC.

Ad esempio, sample.exe

(Avvio possibile se il valore di Impostazioni è Path=C:\Documents and Settings\user\Local Settings\Temp)

## 3. Definire il percorso con la variabile Ambiente

Questa operazione è possibile se il file eseguibile si trova in una cartella specificata dai parametri ambiente delle Impostazioni ambiente su un dispositivo della Serie IPC.

Ad esempio, %TEMP%\sample.exe

(Avvio possibile se il valore del Parametro ambiente è specificato come TEMP=C:\Documents and Settings\user\Local Settings\Temp)

**Espressione esempio 1:**

Autorizza istanze multiple (avviare Blocco note e visualizzare Readme.txt)

```
Exec_Process ("C:\WINDOWS\SYSTEM32\notepad.exe","D:\TEMP\Readme.txt", "",0)
```

```
Exec_Process ("%SystemFolder%\notepad.exe","D:\TEMP\Readme.txt", "",1)
```

**Espressione esempio 2:**

Non consentire istanze multiple:

Word parziali (avviare Blocco note e visualizzare Readme.txt)Exec\_Process

```
("C:\WINDOWS\SYSTEM32\notepad.exe","D:\TEMP\Readme.txt","Readme",0)
```

**Espressione esempio 3:**

Non consentire istanze multiple: Solo word intere (avviare Blocco note e visualizzare Readme.txt)

```
Exec_Process
```

```
("C:\WINDOWS\SYSTEM32\notepad.exe","D:\TEMP\Readme.txt","Readme.txt - Notepad",1)
```

**Espressione esempio 4:**

Non consentire istanze multiple: Word parziali (avviare Blocco note)

```
Exec_Process ("C:\WINDOWS\SYSTEM32\notepad.exe", "", "Notepad",0)
```

**Espressione esempio 5:**

Nessun parametro (avviare Blocco note)

```
Exec_Process ("C:\WINDOWS\SYSTEM32\notepad.exe", "", "",0)
```

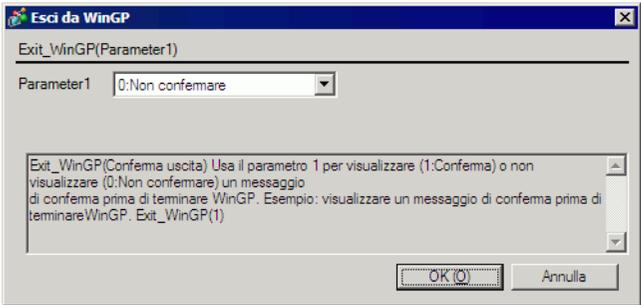
**Espressione esempio 6:**

Parametro multiplo (avviare sample.exe)

```
Exec_Process ("C:\WINDOWS\SYSTEM32\sample.exe", "/v /a/s", "", 1)
```

**■ Esci da WinGP**

Questa funzione è operativa solo sui modelli della Serie IPC.

Elemento	Descrizione
Riepilogo	Uscire da WinGP. All'uscita è possibile visualizzare un messaggio di riconoscimento.
Format	<p>Exit_WinGP(Parameter1)</p>  <p>Parametro 1 Nome cartella: Selezionare "0: Non confermare" o "1: Conferma".</p>

**NOTA**

- Il Parametro1 richiede l'inserimento di testo (percorso EXE). Si verifica un errore se non si inserisce del testo.
- Questa funzione non è operativa quando si trasferisce lo script "Esci da WinGP" su modelli non appartenenti alla Serie IPC.

**Espressione esempio:**

Visualizzazione di un messaggio di riconoscimento all'uscita da WinGP.

```
Exit_WinGP(1)
```

## 21.11.8 Espressioni condizionali

Espressioni condizionali	Riepilogo funzioni
<div style="border: 1px solid gray; padding: 5px; width: fit-content;">           Espressione descrizione  <a href="#">if - endif</a>  <a href="#">if - else - endif</a>  <a href="#">loop - endloop</a>  <a href="#">break</a>  <a href="#">return</a> </div>	<p>if - endif</p> <p>☞ " ■ if - endif" (pagina 21-147)</p> <p>Quando la condizione "if", inclusa nelle parentesi "()", è vera, l'espressione che segue l'istruzione "if ()" verrà eseguita.</p>
	<p>if - else - endif</p> <p>☞ " ■ if - else - endif" (pagina 21-147)</p> <p>Quando la condizione "if", inclusa nelle parentesi "()", è vera, l'espressione che segue l'istruzione "if ()" verrà eseguita.</p> <p>Quando la condizione è falsa, l'espressione dopo "else" verrà eseguita.</p>
	<p>loop - endloop</p> <p>☞ " ■ loop - endloop" (pagina 21-148)</p> <p>L'elaborazione Loop viene ripetuta secondo il numero di volte memorizzato negli indirizzi temporanei designati tra le parentesi "()" che seguono "loop"</p>
	<p>break</p> <p>☞ " ■ break" (pagina 21-151)</p> <p>Arresta l'operazione loop durante l'esecuzione dell'equazione loop ().</p>
	<p>return</p> <p>☞ " ■ return" (pagina 21-151)</p> <p>Riesegue dall'inizio.</p> <p>Può essere utilizzata solo in uno Script esteso.</p>

### ■ if - endif

Quando la condizione "if", inclusa nelle parentesi "()", è vera, l'espressione che segue l'istruzione "if ()" verrà eseguita.

#### NOTA

- Il carattere Assegna "=" non può essere utilizzato in un'espressione condizionale.

### ■ if - else - endif

Quando la condizione "if", inclusa nelle parentesi "()", è vera, l'espressione che segue l'istruzione "if ()" verrà eseguita. Quando la condizione è falsa, l'espressione dopo "else" verrà eseguita.

#### NOTA

- Il carattere Assegna "=" non può essere utilizzato in un'espressione condizionale.

## ■ loop - endloop

L'elaborazione Loop viene ripetuta secondo il numero di volte memorizzato negli indirizzi temporanei designati tra le parentesi “( )” che seguono “loop”

### Loop infinito

Il loop è infinito quando nelle parentesi ( ) del loop non sono contenute espressioni.

I loop infiniti possono essere usati negli script estesi.

### Espressione esempio:

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

#### NOTA

- Il formato loop ( ) è il seguente:

Ad esempio:

```
loop (numero di loop)// Definisce l'Indirizzo temporaneo in cui è
memorizzato il numero dei loop.
```

```
{
```

```
Equazione Azione
```

```
break // Usare per uscire completamente dalla parte del loop
(facoltativo)
```

```
} endloop // Definisce la fine del loop
```

- Solo un Indirizzo word temporaneo può essere inserito nelle parentesi. (Ad esempio, loop ([t:000]))
- “loop ( )” non si può utilizzare per un'equazione trigger.
- Il valore dell'indirizzo word temporaneo usato per definire il numero di loop diminuisce ad ogni loop. Quando il valore diventa 0, l'operazione loop termina. Se il valore dell'indirizzo word temporaneo definito per il numero di loop viene modificato, il loop potrebbe diventare infinito. L'Indirizzo word temporaneo usato viene designato come Globale. Pertanto, l'uso temporaneo dell'indirizzo word per altri fini potrebbe creare un loop infinito.
- Fino al completamento di un'operazione loop, le visualizzazioni schermo di Parti ecc. non saranno aggiornate.

Continua

**NOTA**

- Il loop ( ) può anche essere annidato. Quando è annidato, il loop più interno ( ) verrà saltato mediante il comando "break".

```
loop ([t:0000]) // loop1
{
  loop ([t:0001]) // loop2
  {
    break // Uscita dal loop2
  }endloop
}endloop

break // Uscita dal loop1
}endloop
```

- Se l'operazione loop termina senza utilizzare il comando di uscita, il valore dell'Indirizzo word temporaneo diventa 0.
- L'intervallo disponibile per il valore dell'Indirizzo word temporaneo differirà secondo il formato dei dati (Bin, BCD), la lunghezza dei bit e il codice +/- utilizzati. Se il codice +/- è stato designato e l'Indirizzo word temporaneo diventa un valore negativo, la condizione viene giudicata all'inizio del loop e dell'arresto dell'elaborazione dei loop.
- L'uso di "loop" o "break" come nome per una funzione D-Script causerà un errore.

Continua

**NOTA**

- NON utilizzare un dispositivo PLC nella formula loop. Usare, invece, un indirizzo area utente dell'area LS interna dell'unità di visualizzazione, oppure un Indirizzo word temporaneo. Ad esempio, la seguente descrizione esegue la scrittura di dati nel PLC molte volte in un breve periodo (100 volte, nell'esempio seguente). Questo potrà causare un errore di sistema, poiché l'elaborazione della comunicazione (il tempo necessario per scrivere nel PLC) non può essere eseguita a questa velocità.

Ad esempio:

```
[t:0000] = 100 // 100 loop
loop ([t:0000])
{
  [w:[PLC1]D0200] = [w:[#INTERNAL]LS0100] // Write to D0200
  [w:[#INTERNAL]LS0100] = // Increment LS0100
  [w:[#INTERNAL]LS0100] + 1
}endloop
```

Modificare come segue:

```
[t:0000] = 100 // 100 loop
loop ([t:0000])
{
  [w:[#INTERNAL]LS0200] = // Write to D0200
  [w:[#INTERNAL]LS0100]
  [w:[#INTERNAL]LS0100] = // Increment LS0100
  [w:[#INTERNAL]LS0100] + 1
}endloop
[w:[PLC1]D0200]=[w:[#INTERNAL]LS0200] //LS0200 contents, write
into D0200
```

## ■ break

Esiste un'operazione loop nel mezzo dell'operazione loop ( ).

### NOTA

- Il comando "break" si può usare solo nella sezione { } del loop ( ).
- Gli script non funzioneranno in modo corretto se si usa il comando "break" nelle espressioni { }.

## ■ return

Quando la "Funzione definita dall'utente" include "return", l'elaborazione della Funzione è terminata e il controllo ritorna a chi ha richiamato la Funzione.

Quando Esecuzione (Funzione principale) include "return"

L'elaborazione della funzione principale viene al momento interrotta, e riprenderà dall'inizio della funzione principale.

### NOTA

- Il carattere Assegna "=" non può essere utilizzato in un'espressione condizionale.

### Espressione esempio:

```
[w:[#INTERNAL]LS0100]=[w:[#INTERNAL]LS0200]>> 8) & 0xFF
if ([w:[#INTERNAL]LS0100]==0) // Quando LS0100 è "0", l'elaborazione cessa
{
    set([b:[#INTERNAL]LS005000]) // Imposta l'indirizzo bit per la visualizzazione
                                degli errori
    return //Fine
}
endif
```

## 21.11.9 Confronto

Confronto	Riepilogo funzioni
Confronto <a href="#">AND logico (AND)</a> <a href="#">OR logico (OR)</a> <a href="#">Negazione (not)</a> <a href="#">inferiore a (&lt;)</a> <a href="#">inferiore o uguale a (&lt;=)</a> <a href="#">non uguale a (&lt;&gt;)</a> <a href="#">più di (&gt;)</a> <a href="#">superiore o uguale a (&gt;=)</a> <a href="#">Equivalente (==)</a>	AND logico (and)  " ■ AND logico (and)" (pagina 21-152) N1 AND N2: Vero se N1 e N2 si trovano entrambi su ON.
	OR logico (or)  " ■ OR logico (or)" (pagina 21-152) N1 OR N2: Vero se N1 o N2 si trova su ON.
	Negazione (not)  " ■ Negazione (not)" (pagina 21-152) notN1: diventa 0 se N1 è 1, e 1 se N1 è 0.
	Inferiore a (<)  " ■ Inferiore a (<)" (pagina 21-153) Vero se N1 è superiore a N2 (N1 < N2).
	Inferiore o uguale a (=)  " ■ Inferiore o uguale a (<=)" (pagina 21-153) Vero se N1 è inferiore o uguale a N2 (N1 <= N2).
	Non uguale a (<>)  " ■ Non uguale a (<>)" (pagina 21-153) Vero se N1 non è uguale a N2 (N1 <> N2).
	Superiore a (>)  " ■ Superiore a (>)" (pagina 21-153) Vero se N1 è superiore a N2 (N1 > N2).
	Superiore o uguale a (>=)  " ■ Superiore o uguale a (>=)" (pagina 21-153) Vero se N1 è superiore o uguale a N2 (N1 >= N2).
	Equivalente (==)  " ■ Uguale a (==)" (pagina 21-153) Vero se N1 equivale a N2 (N1 = N2).

### ■ AND logico (and)

Mostra AND sui lati destro e sinistro. Il valore 0 (zero) viene considerato OFF, e gli altri valori ON.

N1 AND N2: Vero se N1 e N2 si trovano entrambi su ON. In ogni altro caso, falso.

### ■ OR logico (or)

Mostra OR sui lati destro e sinistro. Il valore 0 (zero) viene considerato OFF, e gli altri valori ON.

N1 OR N2: Vero se N1 o N2 si trova su ON. In ogni altro caso, falso.

### ■ Negazione (not)

Inverte il valore. 0 (zero) è considerato 1, e gli altri valori 0.

notN1: diventa 0 se N1 è 1, e 1 se N1 è 0.

### ■ Inferiore a (<)

Confronta i dati in due indirizzi word, o i dati in un indirizzo word e in una costante. Vero se N1 è superiore a N2 ( $N1 < N2$ ).

### ■ Inferiore o uguale a (<=)

Confronta i dati in due indirizzi word, o i dati in un indirizzo word e in una costante. Vero se N1 è inferiore o uguale a N2 ( $N1 \leq N2$ ).

### ■ Non uguale a (<>)

Confronta i dati in due indirizzi word, o i dati in un indirizzo word e in una costante. Vero se N1 non è uguale a N2 ( $N1 \neq N2$ ).

### ■ Superiore a (>)

Confronta i dati in due indirizzi word, o i dati in un indirizzo word e in una costante. Vero se N1 è superiore a N2 ( $N1 > N2$ ).

### ■ Superiore o uguale a (>=)

Confronta i dati in due indirizzi word, o i dati in un indirizzo word e in una costante. Vero se N1 è superiore o uguale a N2 ( $N1 \geq N2$ ).

### ■ Uguale a (==)

Confronta i dati in due indirizzi word, o i dati in un indirizzo word e in una costante. Vero se N1 equivale a N2 ( $N1 = N2$ ).

Comando		Ad esempio:
AND logico	and	if ((Operazione) and (Operazione))
OR logico	or	if ((Operazione) or (Operazione))
Negazione	not	if (not (Operazione))
Inferiore a	<	(Termine 1) < (Termine 2)
Inferiore o uguale a	<=	(Termine 1) <= (Termine 2)
Non uguale a	<>	(Termine 1) <> (Termine 2)
Superiore a	>	(Termine 1) > (Termine 2)
Superiore o uguale a	>=	(Termine 1) >= (Termine 2)
Uguale a	==	(Termine 1) == (Termine 2)

## 21.11.10 Operatore

Operatore	Riepilogo funzioni
	<p>Aggiunta (+)   " ■ Addizione (+)" (pagina 21-155)            Somma i dati in due indirizzi word, oppure i dati in un indirizzo word e in una costante.</p>
	<p>Sottrazione (-)   " ■ Sottrazione (-)" (pagina 21-155)            Sottrae i dati in due indirizzi word, o i dati in un indirizzo word e in una costante.</p>
	<p>Modulo (%)   " ■ Modulo (%)" (pagina 21-155)            Individua il resto di una divisione eseguita sui dati in due indirizzi word, o sui dati in un indirizzo word e in una costante.</p>
	<p>Moltiplicazione (*)   " ■ Moltiplicazione (*)" (pagina 21-155)            Moltiplica i dati tra due indirizzi word, o i dati un indirizzo word e i dati di una costante.</p>
<p>Operatore  <a href="#">Aggiunta (+)</a>  <a href="#">Sottrazione (-)</a>  <a href="#">Modulo (%)</a>  <a href="#">Moltiplicazione (*)</a>  <a href="#">Divisione (/)</a>  <a href="#">Assegnazione (=)</a>  <a href="#">Spostamento a sinistra (&lt;&lt;)</a>  <a href="#">MAIUSC di destra (&gt;&gt;)</a>  <a href="#">AND logico dell'operatore sul bit (&amp;)</a>  <a href="#">OR logico dell'operatore sul bit ( )</a>  <a href="#">OR esclusivo dell'operatore sul bit (^)</a>  <a href="#">Complemento di 1 dell'operatore sul bit (~)</a></p>	<p>Divisione (/)   " ■ Divisione (/)" (pagina 21-155)            Confronta i dati in due indirizzi word, o i dati in un indirizzo word e in una costante.</p>
	<p>Assegnazione (=)   " ■ Assegnazione (=)" (pagina 21-155)            Assegnare il valore sulla parte destra alla parte sinistra.</p>
	<p>Spostamento a sinistra (&lt;&lt;)   " ■ Sposta a sinistra (&lt;&lt;)" (pagina 21-155)            Sposta i dati sul lato sinistro verso sinistra in base al numero sul lato destro.</p>
	<p>MAIUSC di destra (&gt;&gt;)   " ■ Sposta a destra (&gt;&gt;)" (pagina 21-156)            Sposta i dati sul lato sinistro verso destra in base al numero sul lato destro.</p>
	<p>Operatore bit AND logico (&amp;)   " ■ Bitwise AND (&amp;)" (pagina 21-156)            Esegue l'AND logico dei dati tra dispositivi word, o tra i dati dei dispositivi word e una costante.</p>
	<p>OR logico dell'operatore sul bit ( )   " ■ Bitwise OR ( )" (pagina 21-156)            Esegue l'OR logico dei dati tra dispositivi word, o tra i dati dei dispositivi word e una costante.</p>

Continua

Operatore	Riepilogo funzioni
Operatore	OR esclusivo dell'operatore sul bit (^)  " ■ Esclusivo Bitwise OR (^)" (pagina 21-156) Esegue l'OR esclusivo dei dati tra dispositivi word, o tra i dati dei dispositivi word e una costante.
	Complemento di 1 dell'operatore sul bit (~)  " ■ Complemento Bitwise 1 (~)" (pagina 21-156) Inverte i bit.

### ■ Addizione (+)

Somma i dati in due indirizzi word, oppure i dati in un indirizzo word e in una costante. Quando i risultati dell'elaborazione sono in eccesso, le cifre saranno troncate.

### ■ Sottrazione (-)

Sottrae i dati in due indirizzi word, o i dati in un indirizzo word e in una costante. Quando i risultati dell'elaborazione sono in eccesso, le cifre saranno troncate.

### ■ Modulo (%)

Individua il resto di una divisione eseguita sui dati in due indirizzi word, o sui dati in un indirizzo word e in una costante. Il risultato dell'operazione può dipendere dal segno dei lati sinistro e destro.

### ■ Moltiplicazione (\*)

Moltiplica i dati tra due indirizzi word, o i dati un indirizzo word e i dati di una costante. Quando i risultati dell'elaborazione sono in eccesso, le cifre saranno troncate.

### ■ Divisione (/)

Confronta i dati in due indirizzi word, o i dati in un indirizzo word e in una costante. I valori frazionali risultanti dall'operazione sono troncati. Quando i risultati dell'elaborazione sono in eccesso, le cifre saranno troncate.

### ■ Assegnazione (=)

Assegnare il valore sulla parte destra alla parte sinistra. Solo gli indirizzi possono essere specificati sulla parte sinistra. Indirizzi e Costanti possono essere descritti sulla destra. Quando i risultati dell'elaborazione sono in eccesso, le cifre saranno troncate.

### ■ Sposta a sinistra (<<)

Sposta i dati sul lato sinistro verso sinistra in base al numero sul lato destro. Questa funzione supporta solo gli spostamenti logici.



**Ordine di precedenza associatività**

La tabella seguente mostra l'ordine di precedenza degli operatori. Se due o più operatori hanno lo stesso l'ordine di precedenza, seguire la direzione indicata dall'associatività.

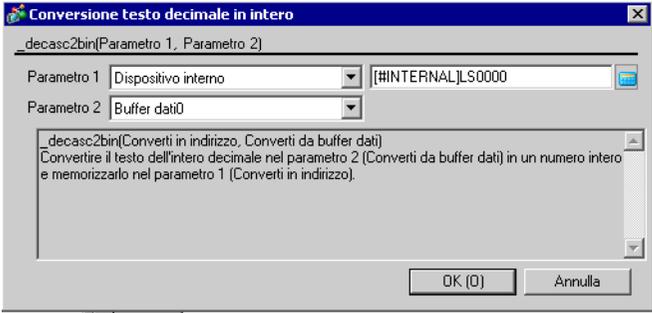
Priorità	Operatore	Associatività
Veloce	()	→
	not ~	←
	* / %	→
	+ -	→
	<< >>	→
	< <= > >=	→
	== <>	→
	& ^	→
	and or	→
	Lento	=

## 21.11.11 Operazione del testo

Le funzioni di Operazione del testo possono essere utilizzate solo negli Script estesi.

Operazione del testo	Riepilogo funzioni
	Conversione testo decimale in intero ☞ " ■ <b>Conversione testo decimale in intero</b> " (pagina 21-159) Questa funzione viene utilizzata per convertire i testi decimali in interi.
	Conversione esadecimale testo in numero intero ☞ " ■ <b>Conversione testo decimale in intero</b> " (pagina 21-161) Questa funzione viene utilizzata per convertire i testi esadecimali in interi.
	Dal dispositivo interno al buffer dati ☞ " ■ <b>Dal dispositivo interno al buffer dati</b> " (pagina 21-163) I dati della stringa memorizzata nel dispositivo interno saranno copiati nel buffer dati.
	Dal buffer dati al dispositivo interno ☞ " ■ <b>Dal buffer dati al dispositivo interno</b> " (pagina 21-165) I dati della stringa memorizzata nel buffer dati saranno copiati nel dispositivo interno.
	Stato ☞ " ■ <b>Stato di errore operazione testo</b> " (pagina 21-167) Memorizza qualsiasi errore verificatosi.
	Conversione stringa decimale del valore numerico ☞ " ■ <b>Conversione del valore numerico in una stringa decimale</b> " (pagina 21-169) Questa funzione viene utilizzata per convertire un numero intero in una stringa decimale.
	Conversione stringa esadecimale del valore numerico ☞ " ■ <b>Conversione del valore numerico in una stringa esadecimale</b> " (pagina 21-170) Questa funzione viene utilizzata per convertire dati binari in una stringa esadecimale.
	Copia stringa parziale ☞ " ■ <b>Testo parziale</b> " (pagina 21-171) I dati saranno recuperati dall'offset specificato della stringa secondo la lunghezza della stringa e memorizzati in un altro buffer dati.
	Impostazioni Testo ☞ " ■ <b>Impostazioni testo</b> " (pagina 21-172) Una stringa fissa viene memorizzata nel buffer dati.
	Lunghezza stringa ☞ " ■ <b>Lunghezza testo</b> " (pagina 21-173) Ottiene la lunghezza della stringa memorizzata.
	Stringa concatenata ☞ " ■ <b>Stringa concatenata</b> " (pagina 21-174) Una stringa carattere o un codice carattere sono concatenati con il buffer di testo.

## ■ Conversione testo decimale in intero

Elemento	Descrizione
Riepilogo	Questa funzione viene utilizzata per convertire una stringa decimale in un numero intero. Convertire il testo decimale del numero intero nel Parametro 2 (Converti dal buffer dati) in un numero intero e memorizzarlo nel Parametro 1 (Converti in indirizzo).
Format	<p><code>_decasc2bin</code> ([Converti in indirizzo], [Converti dal buffer dati])</p>  <p>Parametro 1: Dispositivo interno, Indirizzo temporaneo Parametro 2: Buffer dati</p>

### Espressione esempio 1 (Quando la lunghezza dati è di 16 bit)

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

Il contenuto di “databuf0” è il seguente:

	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULL

I dati sopra sono convertiti nel modo indicato di seguito.

	16 Bit
LS0100	1234

**Espressione esempio 2 (Quando la lunghezza dati è di 32 bit)**

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

Il contenuto di “databuf0” è il seguente:

	8 bit	
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	NULL

I dati sopra sono convertiti nel modo indicato di seguito.

	32 bit
LS0100	12345678
LS0102	

**IMPORTANTE**

- Si verificherà un errore quando la lunghezza bit convertita è superiore alla lunghezza bit dell'editor D-Script.  
Ad esempio, quando la lunghezza bit dello script è pari a 16 bit:  

```
_strset (databuf 0, " 123456") // Quando una cifra decimale di 6 cifre è impostata accidentalmente
```

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

Quando l'espressione sopra viene eseguita, si attiverà l'Errore numero 2 (errore di conversione stringa) dello stato Errore Stringa [e: In ogni caso, il bit ritorna all'inizio della funzione principale quando si verifica un errore. Pertanto, non si può fare riferimento diretto ad altre funzioni dopo l'esecuzione di `_decasc2bin`. (Se il comando arriva mentre viene eseguita una funzione, ritornerà alla linea che ha richiamato la funzione in questione.)
- Si verifica un errore durante la conversione di una stringa di dati contenenti caratteri diversi da quelli da “0” a “9”.  
Ad esempio, quando la lunghezza bit dello script è pari a 16 bit:  

```
_strset (databuf0, "12AB") // Quando una stringa non decimale è impostata accidentalmente
```

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

Quando l'espressione sopra viene eseguita, si attiverà l'Errore numero 2 (errore di conversione stringa) dello stato Errore Stringa [e: In ogni caso, il bit ritorna all'inizio della funzione principale quando si verifica un errore. Pertanto, non si può fare riferimento diretto ad altre funzioni dopo l'esecuzione di `_decasc2bin`. (Se il comando arriva mentre viene eseguita una funzione, ritornerà alla linea che ha richiamato la funzione in questione.)
- L'elaborazione viene terminata quando si verifica un errore e ritorna all'inizio della funzione principale. (Se il comando arriva mentre viene eseguita una funzione, ritornerà alla linea che ha richiamato la funzione in questione.)

## ■ Conversione testo decimale in intero

Elemento	Descrizione
Riepilogo	Questa funzione converte una stringa esadecimale in dati binari. Convertire il testo esadecimale del numero intero nel Parametro 2 (Buffer dati Converti-da) in un numero intero e memorizzarlo nel Parametro 1 (Indirizzo Converti-in).
Format	<p><code>_hexasc2bin ([Indirizzo Converti-in], [Buffer dati Converti-da])</code></p>  <p>Parametro 1: Dispositivo interno, Indirizzo temporaneo Parametro 2: Buffer dati</p>

### Espressione esempio 1 (Quando la lunghezza dati è di 16 bit)

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

Il contenuto di “databuf0” è il seguente:

	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULL

I dati sopra sono convertiti nel modo indicato di seguito.

	16 bit
LS0100	1234h

**Espressione esempio 2 (Quando la lunghezza dati è di 32 bit)**

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

Il contenuto di “databuf0” è il seguente:

	8 bit	
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	NULL

I dati sopra sono convertiti nel modo indicato di seguito.

	32 bit
LS0100	12345678h
LS0102	

**IMPORTANTE**

- Si verifica un errore quando la stringa convertita è superiore a 16 bit o a 32 bit.

Ad esempio, quando la lunghezza bit dello script è pari a 16 bit:

```
_strset (databuf0, "123456")
```

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

Quando l'espressione sopra viene eseguita, si attiverà l'Errore numero 2 (errore di conversione stringa) dello stato Errore Stringa [e:

- Si verifica un errore durante la conversione di una stringa di dati contenenti caratteri diversi da quelli da “0” a “9”, da “A” a “F”, o da “a” a “f”.

Ad esempio, quando la lunghezza bit dello script è pari a 16 bit:

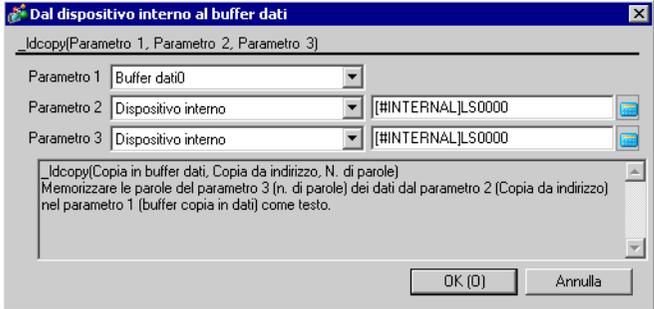
```
_strset (databuf 0, "123G")
```

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

Quando l'espressione sopra viene eseguita, si attiverà l'Errore numero 2 (errore di conversione stringa) dello stato Errore Stringa [e:

- L'elaborazione viene terminata quando si verifica un errore e ritorna all'inizio della funzione principale. (Se il comando arriva mentre viene eseguita una funzione, ritornerà alla linea che ha richiamato la funzione in questione.)

■ Dal dispositivo interno al buffer dati

Elemento	Descrizione
Riepilogo	I dati della stringa memorizzata nell'area LS area vengono copiati nel buffer dati secondo il numero di stringhe in un trasferimento byte-per-byte. Memorizzare il Parametro 3 (Word), le word di dati provenienti dal Parametro 2 (Indirizzo Copia-da) nel Parametro 1 (Copia-in buffer dati) sotto forma di testo.
Format	<p><code>_ldcopy (Copia-in Buffer dati, [Indirizzo Copia-da], Word)</code></p>  <p>Parametro 1: Buffer dati                      Parametro 2: Dispositivo interno                      Parametro 3: Valore numero intero, Dispositivo interno, Indirizzo temporaneo (l'intervallo valido per il Parametro 3 va da 1 a 1024).</p>

**Espressione esempio 1:**

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

	16 bit
LS0100	31h
LS0101	32h
LS0102	33h
LS0103	34h

I dati nelle aree da LS0100 a LS0103 sono scritti nei 4 byte del buffer dati in sequenza, iniziando da "databuf0". L'area LS viene letta in ognuno dei byte (i bit più bassi).

	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULL

**IMPORTANTE**

- Il byte 1 nella parte bassa dell'area LS viene estrapolato e la quantità specificata di dati viene scritta nel buffer dati.
- Il valore massimo che può essere assegnato al Parametro 3 è 1024. Quando si imposta un valore che supera il limite, verrà attivato l'Errore numero 1 (sovraccarico stringa) dello stato di errore Stringa [e: STR\_ERR\_STAT].
- Anche quando esistono dati nel byte superiore del dispositivo intero, saranno letti solo i dati dei byte di fondo.
- L'elaborazione viene terminata quando si verifica un errore e ritorna all'inizio della funzione principale. (Se il comando arriva mentre viene eseguita una funzione, ritornerà alla linea che ha richiamato la funzione in questione.)

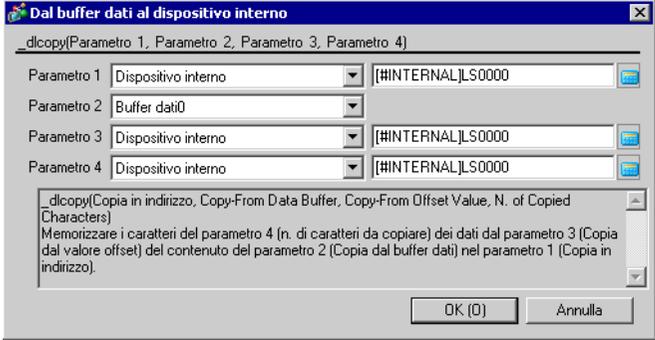
`_ldcopy (databuf0, [w:[#INTERNAL]LS0100], 4)`

	16 bit
LS0100	3132h
LS0101	3334h
LS0102	3536h
LS0103	3738h

Quando i dati sono memorizzati come illustrato sopra, i dati dal byte di fondo vengono letti e scritti nel buffer dati.

	8 bit	
databuf0[0]	32h	'2'
databuf0[1]	34h	'4'
databuf0[2]	36h	'6'
databuf0[3]	38h	'8'
databuf0[4]	00h	NULL

### ■ Dal buffer dati al dispositivo interno

Elemento	Descrizione
Riepilogo	Ciascun byte dei dati della stringa memorizzato nell'offset del buffer dati viene copiato nell'area LS in base al numero di stringhe. Memorizza il Parametro 4 (Caratteri da copiare), caratteri di dati provenienti dal Parametro 3 (Valore di offset Copia-da), del contenuto del Parametro 2 (Copia-da buffer dati) nel Parametro 1 (Copia-in indirizzo).
Format	<p><code>_dlcopy</code> ([Copia-a indirizzo], Copia-dal buffer dati, Copia da- Valore di offset, Numero di caratteri copiati)</p>  <p>Parametro 1: Dispositivo interno                      Parametro 2: Buffer dati                      Parametro 3: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 3 va da 1 a 1024).                      Parametro 4: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 4 va da 1 a 1024).</p>

### Espressione esempio 1:

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

	8 bit	
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'

4 byte di dati recuperati dall'"offset 2" di "databuf0" sono scritti nelle aree da LS0100 a LS0103. I dati sono scritti nell'area LS in unità di 1 byte.

	16 bit
LS0100	33h
LS0101	34h
LS0102	35h
LS0103	36h

**IMPORTANTE**

- 1 byte di dati viene estrapolato dal buffer dati e scritto nell'area LS. Questo significa che saranno usati solo gli 8 bit più bassi (1 byte) dell'area LS. Gli 8 bit significativi (1 byte) saranno azzerati con "0".
  - Quando il valore specificato [valore di offset di origine + numero di caratteri da copiare] supera le dimensioni del buffer verrà attivato l'Errore numero 3 (errore di estrazione stringa) dallo stato di errore stringa [e: STR\_ERR\_STAT].
  - L'elaborazione viene terminata quando si verifica un errore e ritorna all'inizio della funzione principale. (Se il comando arriva mentre viene eseguita una funzione, ritornerà alla linea che ha richiamato la funzione in questione.)
-

## ■ Stato di errore operazione testo

Quando si verifica un errore durante l'esecuzione dell'operazione di testo, viene impostato lo stato di errore nello Stato Errore operazione testo [e: STR\_ERR\_STAT]. "0" in [e: STR\_ERR\_STAT] indica la condizione normale, mentre valori diversi da "0" memorizzati in [e: STR\_ERR\_STAT] indicano stati di errore. L'errore più recente è memorizzato nello Stato Errore operazione testo [e: STR\_ERR\_STAT]. Lo Stato Errore operazione testo può essere configurato con [Operazione porta SIO/Impostazioni etichetta] nel menu Toolbox D-Script. La tabella seguente elenca gli errori di operazione testo.

Numero errore	Messaggio di errore	Descrizione
0	Normale	Nessun errore
1	Sovraccarico testo	Una stringa di almeno 256 byte viene inclusa direttamente nell'argomento per le seguenti funzioni: <code>_strset()</code> , <code>_strlen()</code> , <code>_strcat()</code> , <code>_strmid()</code> , e <code>IO_READ_WAIT()</code> . In alternativa, viene creata una stringa che superi le dimensioni del buffer dati durante l'esecuzione della funzione <code>_strcat()</code> o della funzione <code>_ldcopy()</code> . Ad esempio: <code>_strcat(databuf0, databuf1)</code> La funzione sopra indicata viene eseguita quando una stringa di 1020 byte viene memorizzata in <code>databuf1</code> e una stringa di 60 byte viene memorizzata in <code>databuf 1</code> (una stringa che superi i 1024 byte, dimensioni del buffer dati, risulterà in stato di errore).
2	Errore conversione stringa	Un codice carattere non valido viene assegnato alla funzione <code>_hexasc2bin()</code> o alla funzione <code>_decasc2bin()</code> . Ad esempio: Un codice carattere diverso da quelli da "0" a "9", da "A" a "F", o da "a" a "f" viene incluso nel secondo argomento di <code>_hexasc2bin()</code> .
3	Errore recupero stringa	Viene tentato il recupero di un carattere stringa più lungo di quello specificato nella funzione <code>"_strmid()"</code> . In alternativa, viene designato un valore di offset superiore a quello della stringa specificata. Ad esempio: <code>_strmid(databuf0, "12345678", 2, 8)</code> Viene tentato il recupero di una stringa di 8 caratteri dall'offset 2.

Lo Stato Errore di controllo stringa non può essere utilizzato con i D-Script e i D-Script globali. Se viene estrapolato accidentalmente, sarà caricato uno "0".

Viene memorizzato nello stato di errore durante l'esecuzione di ciascuna funzione.

Per verificare l'errore [e: STR\_ERR\_STAT], scrivere le seguenti istruzioni. Si può confermare l'errore con la seguente espressione.

### Espressione esempio:

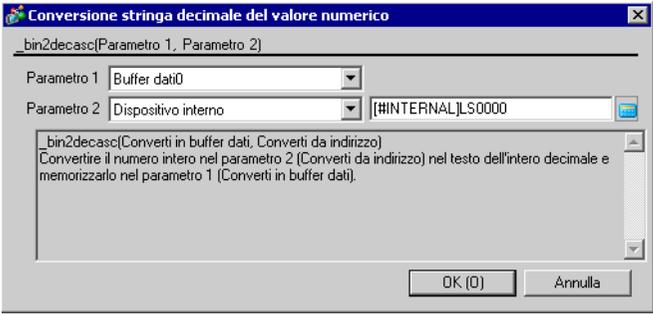
```
if ([e:STR_ERR_STAT] <> 0)           // Controlla lo stato degli errori.
{
    set ([b:[#INTERNAL]LS005000])    // Imposta il bit sulla spia di visualizzazione
                                     degli errori
}
endif
```

---

**IMPORTANTE**

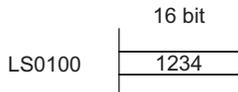
- L'elaborazione viene terminata quando si verifica un errore e ritorna all'inizio della funzione principale. (Se il comando arriva mentre viene eseguita una funzione, ritornerà alla linea che ha richiamato la funzione in questione.)
-

## ■ Conversione del valore numerico in una stringa decimale

Elemento	Descrizione
Riepilogo	Questa funzione viene utilizzata per convertire un numero intero in una stringa decimale. Convertire il numero intero nel Parametro 2 (Indirizzo Converti-da) in un testo intero di un decimale e lo memorizza nel Parametro 1 (Buffer dati Converti-in).
Format	<p><code>_bin2decasc(Indirizzo risultato conversione, Buffer origine conversione)</code></p>  <p>Parametro 1: Buffer dati Parametro 2: Dispositivo interno, Indirizzo temporaneo</p>

### Espressione esempio 1 (Quando la lunghezza dati è di 16 bit)

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

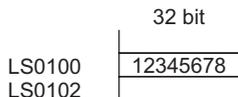


I dati sopra sono convertiti nel modo indicato di seguito. Nota: è stato aggiunto “NULL (0x00)”.

	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULL

### Espressione esempio 2 (Quando la lunghezza dati è di 32 bit)

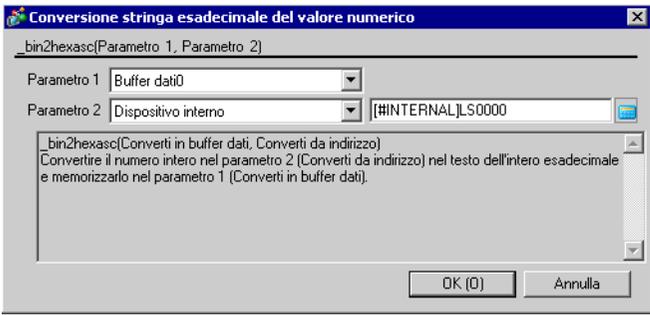
`_bin2decasc (databuf0, [w:[#INTERNAL]LS0100])`



I dati sopra sono convertiti nel modo indicato di seguito.

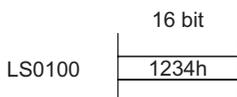
	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	35h	'5'
databuf0[5]	36h	'6'
databuf0[6]	37h	'7'
databuf0[7]	38h	'8'
databuf0[8]	00h	NULL

## ■ Conversione del valore numerico in una stringa esadecimale

Elemento	Descrizione
Riepilogo	Questa funzione viene utilizzata per convertire dati binari in una stringa esadecimale. Convertire il numero intero nel Parametro 2 (Indirizzo Converti-da) in un testo intero di un esadecimale e lo memorizza nel Parametro 1 (Buffer dati Converti-in).
Format	<p><code>_bin2hexasc (Buffer dati Converti-in, [Indirizzo Converti-da])</code></p>  <p>Parametro 1: Buffer dati Parametro 2: Dispositivo interno, Indirizzo temporaneo</p>

### Espressione esempio 1 (Quando la lunghezza dati è di 16 bit)

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

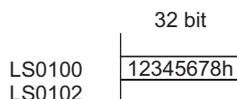


I dati sopra sono convertiti nel modo indicato di seguito. Nota: è stato aggiunto “NULL (0x00)”.

	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULL

### Espressione esempio 2 (Quando la lunghezza dati è di 32 bit)

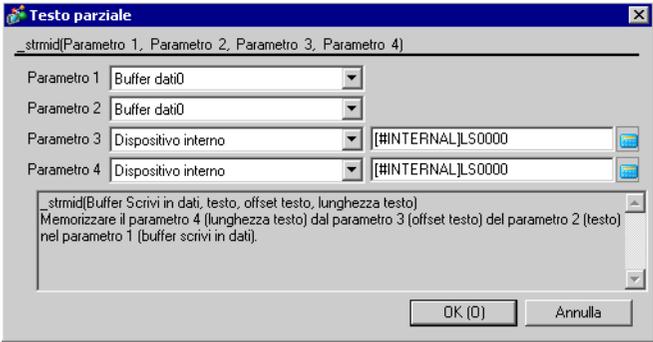
`_bin2hexasc (databuf0, [w:[#INTERNAL]LS0100])`



I dati sopra sono convertiti nel modo indicato di seguito.

	8 bit	
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	NULL

## ■ Testo parziale

Elemento	Descrizione
Riepilogo	I dati saranno recuperati dall'offset specificato della stringa secondo la lunghezza della stringa e memorizzati in un altro buffer dati. Memorizzare il Parametro 4 (Lunghezza testo) dal Parametro 3 (Offset testo) del Parametro 2 (Testo) nel Parametro 1 (Buffer dati Scrivi-in).
Format	<p>_strmid (Buffer dati Scrivi-in, Testo, Offset testo, Lunghezza testo)</p>  <p>Parametro 1: Buffer dati  Parametro 2: Stringa, Buffer dati  Parametro 3: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 3 va da 0 a 1024).  Parametro 4: Valore numerico, Dispositivo interno, Indirizzo temporaneo (L'intervallo valido per il Parametro 4 va da 0 a 1024).</p>

### Espressione esempio:

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

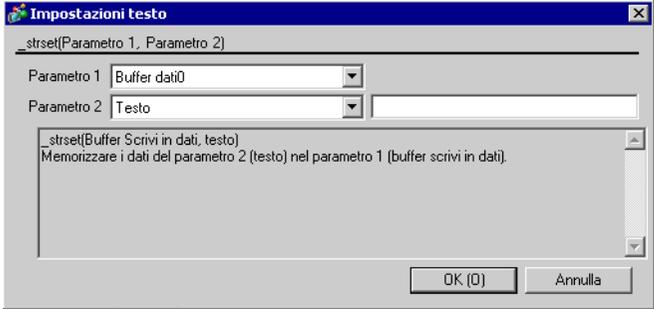
4 byte di dati recuperati dall'offset 2 della stringa "12345678" sono memorizzati in "databuf0".

	8 bit	
databuf0[0]	33h	'3'
databuf0[1]	34h	'4'
databuf0[2]	35h	'5'
databuf0[3]	36h	'6'
databuf0[4]	00h	NULL

### IMPORTANTE

- Quando si tenta di recuperare una stringa più lunga di quella specificata nella funzione "strmid ( )", o quando si specifica un valore di offset superiore a quello della stringa specificata, si attiva l'Errore numero 3 (errore di estrazione stringa) dello stato di errore stringa [e: STR\_ERR\_STAT].
- L'elaborazione viene terminata quando si verifica un errore e ritorna all'inizio della funzione principale. (Se il comando arriva mentre viene eseguita una funzione, ritornerà alla linea che ha richiamato la funzione in questione.)

## ■ Impostazioni testo

Elemento	Descrizione
Riepilogo	Una stringa fissa viene memorizzata nel buffer dati. Memorizza i dati del Parametro 2 (Testo) nel Parametro 1(Buffer dati Scrivi-in).
Format	<p><code>_strset</code>(Scrivi destinazione buffer, Stringa)</p>  <p>Parametro 1: Buffer dati  Parametro 2: Testo, Valore numerico (Codice testo) (L'intervallo valido per il Parametro 2 è 0 e da 1 a 255.)</p>

### Espressione esempio:

`_strset (databuf0, "ABCD")`

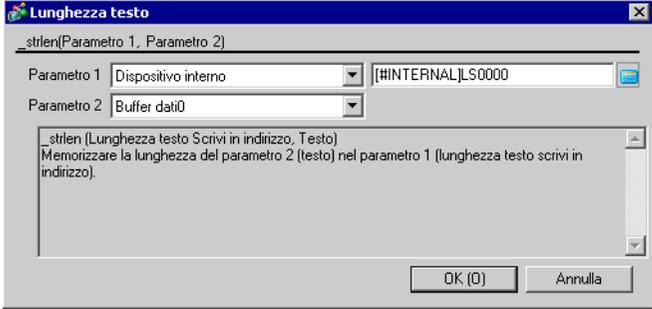
Una stringa fissa viene memorizzata nel buffer dati.

	8 bit	
databuf0[0]	41h	'A'
databuf1[1]	42h	'B'
databuf2[2]	43h	'C'
databuf3[3]	44h	'D'
databuf4[4]	00h	NULL

### IMPORTANTE

- Si possono specificare fino a 255 caratteri (byte). Per creare stringhe più lunghe di questo limite, memorizzare la stringa in un altri buffer o concatenare le stringhe con la funzione concatenante delle stringhe (`_strcat`).
- Per azzerare il buffer di dati, creare una stringa vuota.  
Ad esempio: `_strset (databuf0, "")`  
`_strset (databuf0, 0)`

## ■ Lunghezza testo

Elemento	Descrizione
Riepilogo	Ottiene la lunghezza della stringa memorizzata. Memorizza la lunghezza del Parametro 2 (Testo) nel Parametro 1 (Lunghezza del testo dell'Indirizzo Scrivi-in). (Il carattere NULL non è incluso.)
Format	<p><code>_strlen</code> (Indirizzo destinazione, Stringa)</p>  <p>Parametro 1: Dispositivo interno, Indirizzo temporaneo Parametro 2: Stringa, Buffer dati</p>

### Espressione esempio 1:

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

Quando si esegue l'istruzione sopra, la lunghezza della stringa viene scritta in LS0100 come illustrato di seguito.

LS0100	4
--------	---

### Espressione esempio 2:

```
_strlen ([t:0000], databuf0)
```

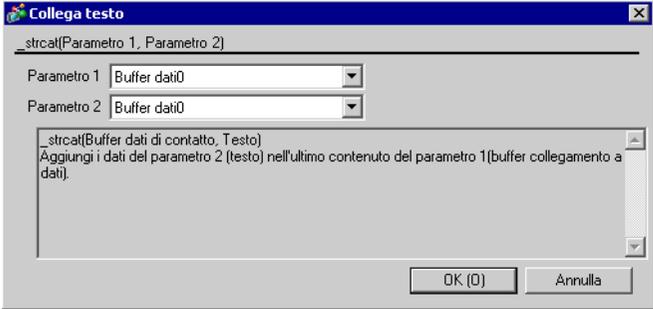
Il contenuto di "databuf0" è il seguente:

	8 bit	
databuf0[0]	31h	'1'
databuf1[1]	32h	'2'
databuf2[2]	33h	'3'
databuf3[3]	34h	'4'
databuf4[4]	00h	NULL

Quando si esegue l'istruzione sopra, la lunghezza della stringa viene scritta in [t: 0000] come illustrato di seguito.

t0000	4
-------	---

## ■ Stringa concatenata

Elemento	Descrizione
Riepilogo	Una stringa carattere o un codice carattere sono concatenati con il buffer di testo. Somma i dati del Parametro 2 (Testo) all'ultimo dei contenuti del Parametro 1(Buffer dati Contatto).
Format	<p><code>_strcat(Buffer dati stringa, Stringa)</code></p>  <p>Parametro 1: Buffer dati Parametro 2: Testo, Valore numerico (Codice testo) Buffer dati (L'intervallo valido per il Parametro 2 è 0 e da 1 a 255.)</p>

### Espressione esempio 1:

`_strcat (databuf0, "ABCD")`

	8 bit	
databuf0[0]	31h	'1'
databuf1[1]	32h	'2'
databuf2[2]	33h	'3'
databuf3[3]	34h	'4'
databuf4[4]	00h	NULL

Quando "ABCD" è concatenata come descritto sopra, il risultato è il seguente. Nota: è stato aggiunto "NULL (0x00)".

	8 bit	
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	NULL

#### IMPORTANTE

- Si possono specificare fino a 255 caratteri (byte).
- Se si imposta una stringa vuota per il valore numerico 0 nel Parametro 2, il buffer dati del Parametro 1 non subirà variazioni. Ad esempio: `_strcat (databuf0,"")`  
`_strcat (databuf0,0)`

## 21.11.12 Esempio operazione

### ■ Esempi di operazioni logiche

Di seguito vengono illustrati esempi di operazioni logiche.

◆ **( ( 100 > 99 ) and ( 200 <> 100 ) )**

Risultato: Uso

◆ **( ( 100 > 99 ) and ( 200 <> 200 ) )**

Risultato: Disattivato

◆ **( ( 100 > 99 ) or ( 200 <> 200 ) )**

Risultato: Uso

◆ **( ( 100 < 99 ) or ( 200 <> 200 ) )**

Risultato: Disattivato

◆ **not ( 100 > 99 )**

Risultato: Disattivato

◆ **not ( 100 < 99 )**

Risultato: Uso

◆ **[ w:[PLC1]D200 ] < 10**

Risultato: Vero se D200 è inferiore a 10.

◆ **not [ w:[PLC1]D200 ]**

Risultato: Vero se D200 è 0.

◆ **( [ w:[PLC1]D200 ] == 2 ) or ( [ w:[PLC1]D200 ] == 5 )**

Risultato: Vero se D200 è 2 o 5.

◆ **( [ w:[PLC1]D200 ] < 5 ) and ( [ w:[PLC1]D300 ] < 8 )**

Risultato: Vero se D200 è inferiore a 5 e D300 è inferiore a 8.

◆ **[ w:[PLC1]D200 ] < 10**

Risultato: Vero se D200 è inferiore a 10.

◆ **not [ w:[PLC1]D200 ]**

Risultato: Vero se D200 è 0.

◆ **( [ w:[PLC1]D200 ] == 2 ) or ( [ w:[PLC1]D200 ] == 5 )**

Risultato: Vero se D200 è 2 o 5.

◆ **( [ w:[PLC1]D200 ] < 5 ) and ( [ w:[PLC1]D300 ] < 8 )**

Risultato: Vero se D200 è inferiore a 5 e D300 è inferiore a 8.

## ■ Esempi di operazioni bit

Di seguito vengono illustrati esempi di operazioni di bit.

### ◆ [ w:[PLC1]D200 ] << 4

Risultato: i dati in D200 sono spostati a sinistra di 4 bit.

### ◆ [ w:[PLC1]D200 ] >> 4

Risultato: Risultato I dati in D200 sono spostati a sinistra di 4 bit.

### ◆ 12(0000Ch) è memorizzata in D301, usando il formato BIN.

[ w:[PLC1]D200 ] = [ w:[PLC1]D300 ] >> [ w:[PLC1]D301 ]

Risultato: i dati in D300 sono spostati a sinistra di 12 bit ed assegnati a D200.

### ◆ [ w:[PLC1]D200 ] << 4

Risultato: i dati in D200 sono spostati a sinistra di 4 bit.

### ◆ [ w:[PLC1]D200 ] >> 4

Risultato: Risultato I dati in D200 sono spostati a sinistra di 4 bit.

### ◆ 12(0000Ch) è memorizzata in D310, usando il formato BIN.

[ w:[PLC1]D200 ] = [ w:[PLC1]D300 ] >> [ w:[PLC1]D301 ]

Risultato: i dati in D300 sono spostati a sinistra di 12 bit ed assegnati a D200.

### ◆ Bitwise AND

0 & 0	Risultato: 0
0 & 1	Risultato: 0
1 & 1	Risultato: 1
0x1234 & 0xF0F0	Risultato: 0x1030

### ◆ Bitwise OR

0 ! 0	Risultato: 0
0 ! 1	Risultato: 1
1 ! 1	Risultato: 1
0x1234   0x9999	Risultato: 0x9BBD

### ◆ Bitwise XOR

0 ^ 0	Risultato: 0
0 ^ 1	Risultato: 1
1 ^ 1	Risultato: 0

### ◆ Il complemento di Bitwise 1 (quando il formato dati corrisponde a Bin16 +)

~ 0	Risultato: 0xFFFF
~ 1	Risultato: 0xFFFE

**■ Esempi di calcolo sull'uso della diramazione condizionale****Controllare il flusso del programma usando "if-endif" e "if-else-endif"****◆ if - endif**

```
if (condizione)
{Processo 1}
endif
```

Se la condizione è vera, sarà eseguito il Processo 1. Se è falsa, Processo 1 verrà saltato.

Ad esempio:

```
if ( [ w:[PLC1]D200 ] < 5 )
{
  [ w:[PLC1]D100 ] = 1
}
endif
```

Se i dati in D200 sono inferiori a 5, assegnerà 1 a D100.

**◆ if - else - endif**

```
if (condizione)
{Processo 1}
else
{Processo 2}
endif
```

Se la condizione è vera, eseguirà Processo 1. Se è falsa, eseguirà Processo 2.

Ad esempio:

```
if ( [ w:[PLC1]D200 ] < 5 )
{
  [ w:[PLC1]D100 ] = 1
}
else
{
  [ w:[PLC1]D100 ] = 0
}
endif
```

Se il valore in D200 è inferiore a 5, assegnerà 1 a D100. Altrimenti, assegnerà 0.

## ■ Esempi di calcolo sull'uso dell'indirizzo di offset

Specifiche offset: Esempi di calcolo speciale usando [w:D00100]#[t:0000].

- ◆ I/O Script: 16 bit senza contrassegno, [t:0000]= 65526, l'indirizzo risultante è [w:D00090].

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

i 16 bit di fondo sono validi

- ◆ I/O Script: 16 bit con contrassegno, [t:0000]= -10, l'indirizzo risultante è [w:D00090].

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

i 16 bit di fondo sono validi

- ◆ I/O Script: 32 bit senza contrassegno, [t:0000]= 4294901840, l'indirizzo risultante è [w:D00180].

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

i 16 bit di fondo sono validi

- ◆ I/O Script: 32 bit senza contrassegno, [t:0000]= -65456, l'indirizzo risultante è [w:D00180].

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

i 16 bit di fondo sono validi

### IMPORTANTE

- Gli indirizzi offset sono sempre trattati come 16 bit Bin, indipendentemente dalla lunghezza di bit dello script e dalle impostazioni del tipo di dati. Se i risultati superano i 16 bit (Valore max: 65535), i bit da 0 a 15 saranno considerati e trattati come bit validi, mentre i bit da 16 in avanti saranno scartati.