

21 | Utilisation de scripts

Ce chapitre explique la façon d'utiliser GP-Pro EX pour créer des scripts.

Lisez tout d'abord la section «21.1 Menu de configuration» (page 21-2), puis passez à la page correspondante.

| | | |
|-------|---------------------------------------------------------------|-------|
| 21.1 | Menu de configuration..... | 21-2 |
| 21.2 | Opérations conditionnelles..... | 21-6 |
| 21.3 | Copie des données en blocs..... | 21-12 |
| 21.4 | Affichage d'une alarme lorsqu'une erreur se produit..... | 21-17 |
| 21.5 | Communication avec des périphériques non pris en charge | 21-21 |
| 21.6 | Créer une mosaïque entre deux scripts | 21-38 |
| 21.7 | Création de scripts | 21-42 |
| 21.8 | Configuration de la condition de déclenchement | 21-47 |
| 21.9 | Guide de configuration | 21-54 |
| 21.10 | Restrictions | 21-60 |
| 21.11 | Commandes de programme/Expressions conditionnelles | 21-69 |

21.1 Menu de configuration

Vous pouvez utiliser des D-Scripts pour créer des programmes simples. Vous pouvez effectuer des opérations sur le GP ou établir une communication entre le GP et des périphériques non pris en charge à l'aide de cette fonction.

 **AVERTISSEMENT**

Assurez-vous de ne pas utiliser des D-scripts ou des D-scripts globaux pour contrôler les systèmes qui peuvent causer des actions constituant un danger de mort ou des blessures graves.

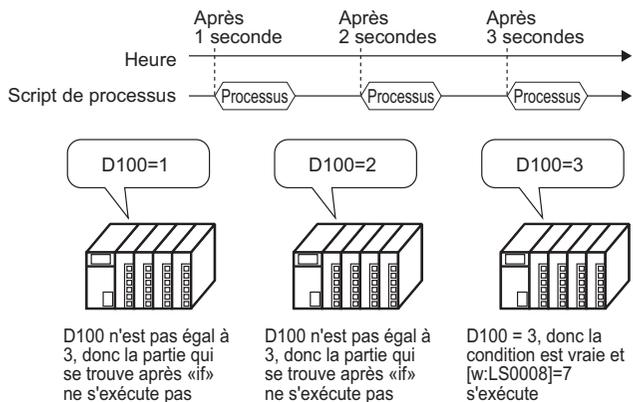
REMARQUE

- Les D-scripts sont configurés dans un écran de base. Cet écran de base examine les conditions pendant son affichage et exécute le script.
- Lorsque le GP est en cours d'exécution, un D-script global s'exécute selon le déclenchement, peu importe l'écran affiché.
- Les scripts étendus devraient être utilisés pour des programmes de communication à haut niveau.
- En plus des scripts, vous pouvez utiliser des programmes logiques pour les applications de contrôle.

 «28.1 Menu de configuration» (page 28-2)

Opérations conditionnelles

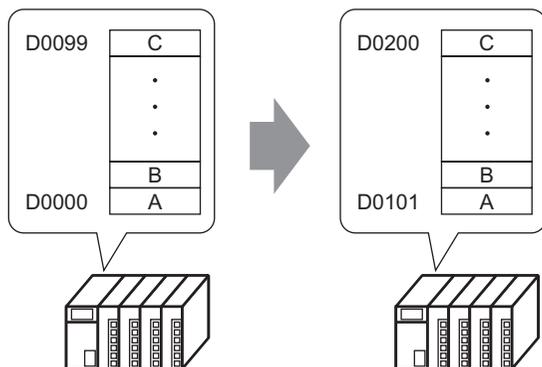
Créez un script qui passe automatiquement à l'écran numéro 7 après 3 secondes.



- ☞ Procédure de configuration (page 21-7)
- ☞ Introduction (page 21-6)

Copie des données en blocs

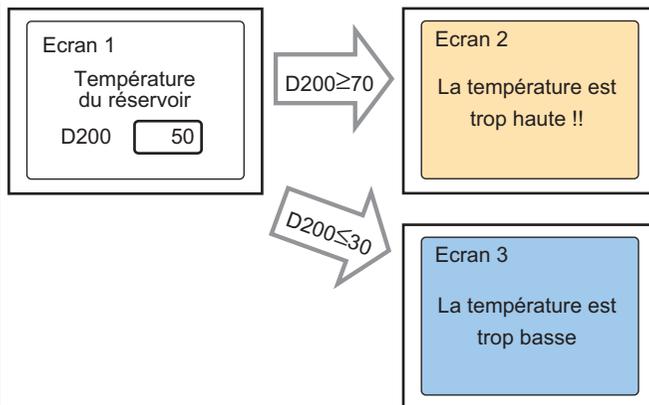
Créez un script qui détecte le front montant (de 0 à 1) de l'adresse de bit M0100 et copie les données stockées dans le périphérique connecté dans une autre adresse.



- ☞ Procédure de configuration (page 21-13)
- ☞ Introduction (page 21-12)

Affichage d'une alarme lorsqu'une erreur se produit

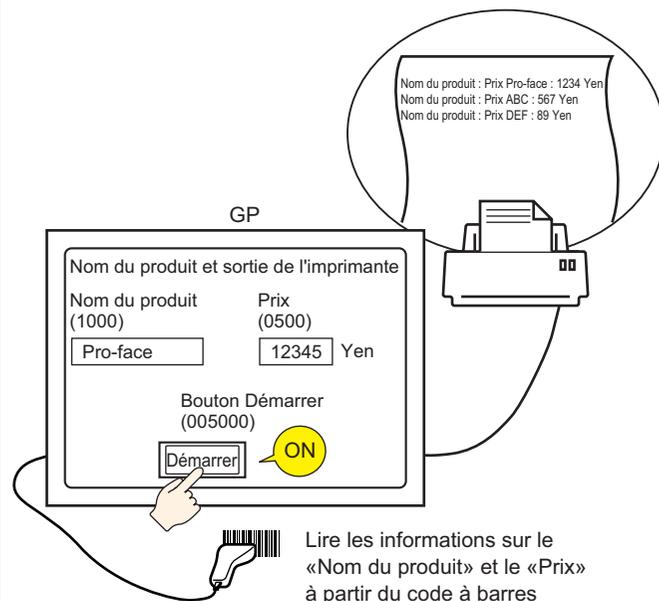
Le système de gestion de températures détecte un bit d'erreur dans le périphérique connecté et affiche des messages d'alarme lorsque l'adresse de stockage de l'information sur la température (D200) monte à 70° C ou plus, ou descend à 30° C ou moins. De plus, ce script compte le nombre d'erreurs détectées.



- ☞ Procédure de configuration (page 21-18)
- ☞ Introduction (page 21-17)

Communication avec des périphériques non pris en charge

Créez un script étendu pour lire des données depuis un code barres connecté au port USB et écrire des données vers une imprimante série connectée au port COM1.



- ☞ Procédure de configuration (page 21-34)
- ☞ Introduction (page 21-21)

Créer une mosaïque entre deux scripts

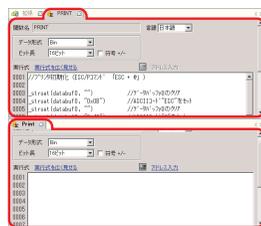
Dans la boîte de dialogue [D-Script], fractionnez l'écran en deux écrans horizontaux ou verticaux.

Cliquer sur les onglets pour basculer les écrans.

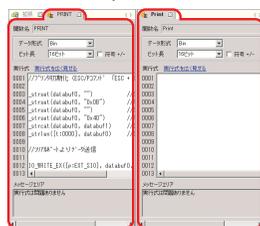


Afficher/Editer deux écrans à côté de l'un l'autre.

Mosaïque horizontale



Mosaïque verticale



- ➡ Procédure d'exploitation (page 21-39)
- ➡ Introduction (page 21-38)

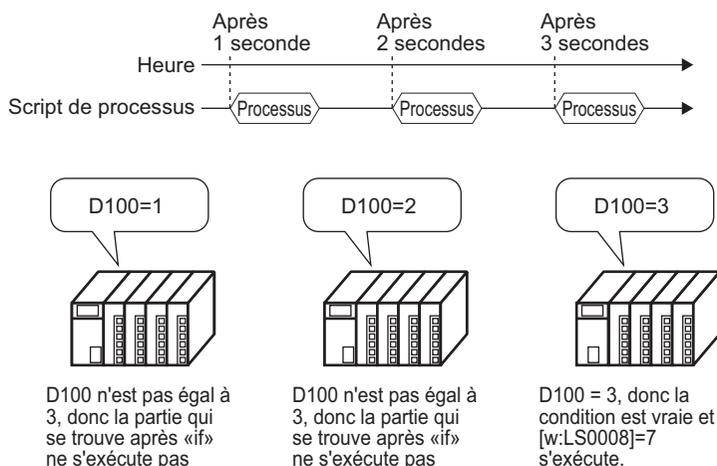
21.2 Opérations conditionnelles

REMARQUE

- Pour en savoir plus, reportez-vous au guide de configuration.
 ➔ «21.9.1 Guide de configuration du D-Script/Outils avancés [D-Script global]» (page 21-54)
- Reportez-vous à ce qui suit pour obtenir plus d'informations sur les commandes disponibles pour les scripts.
 ➔ «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69)

Action

Créez un script qui passe automatiquement à l'écran numéro 7 après 3 secondes.

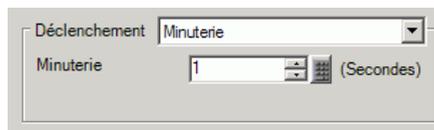


Commandes utilisées

| Commande | Résumé de la fonction |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Affectation (=) | Affecte la valeur du côté droit dans le côté gauche. ➔ «21.11.10 Opérateur» (page 21-156) |
| Addition (+) | Ajoute une constante aux données du périphérique de mot. ➔ «21.11.10 Opérateur» (page 21-156) |
| if () | Lorsqu'une condition devient vraie, le processus qui suit l'instruction «if ()» est exécuté. ➔ «21.11.8 Expressions conditionnelles» (page 21-149) |
| Equivalent (==) | Compare la valeur des côtés droit et gauche. Vrai si le côté gauche est égal au côté droit. ➔ «21.11.9 Comparaison» (page 21-154) |
| LS0008 | Passes au numéro d'écran stocké dans la valeur actuelle. ➔ «A.1.4.2 Zone système» (page A-11) |

Déclenchement

Sélectionnez la minuterie et définissez les [Paramètres de minuterie] sur 1 seconde.



Script terminé

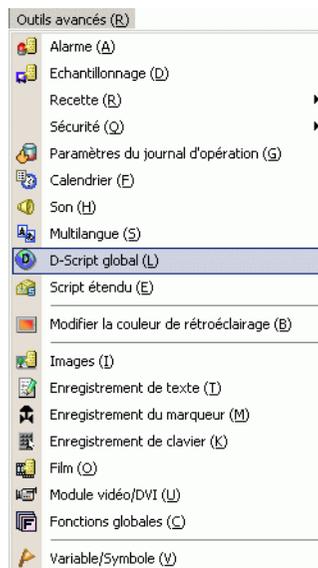
```

Expression d'exécution  Agrandir l'expression d'exécution  Saisie d'adresse
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

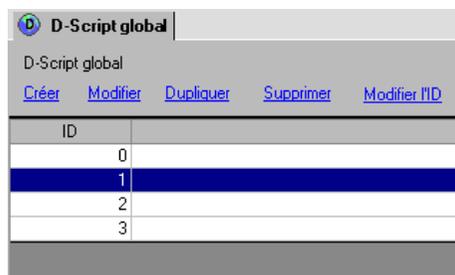
```

Procédure de création

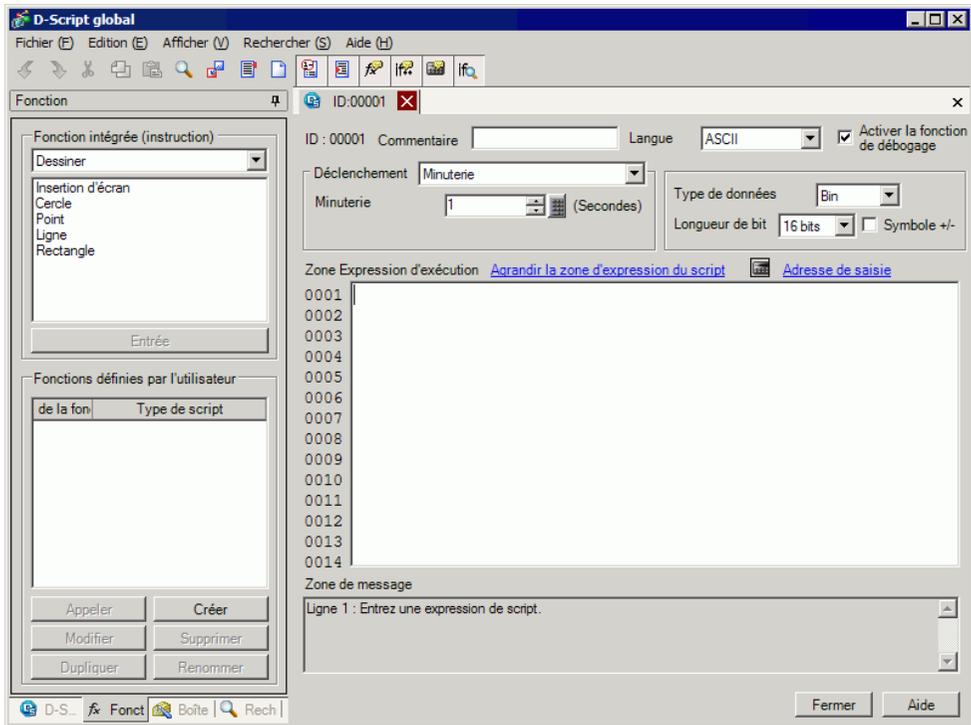
1 Dans le menu [Outils avancés (R)], sélectionnez [D-Script global (L)].



2 Cliquez sur [Créer]. Pour afficher un script existant, sélectionnez le numéro ID et cliquez sur [Modifier], ou cliquez deux fois sur la ligne du numéro ID.

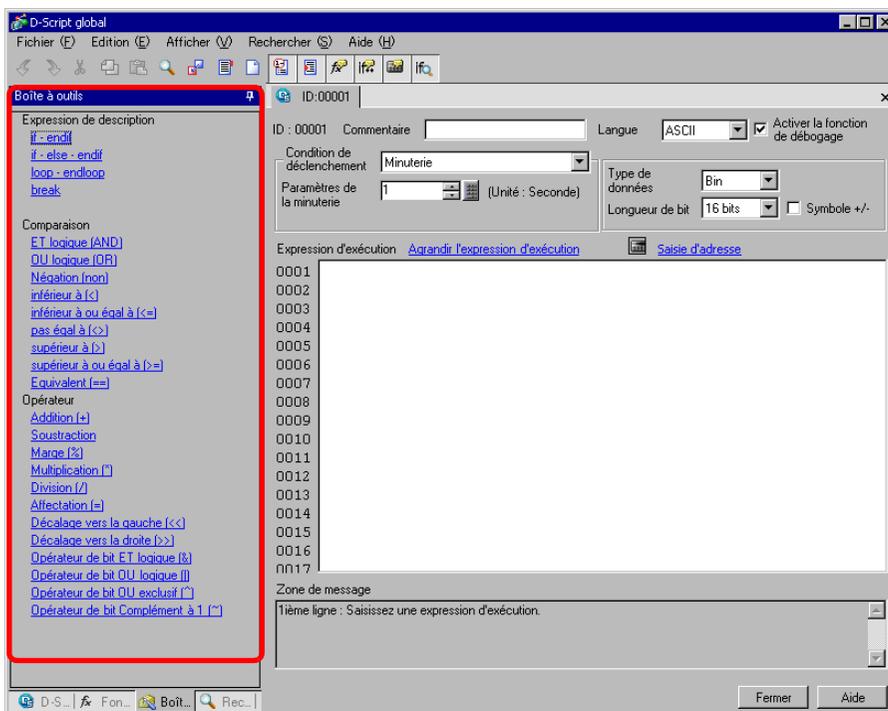


3 La boîte de dialogue [D-Script] s'affiche.



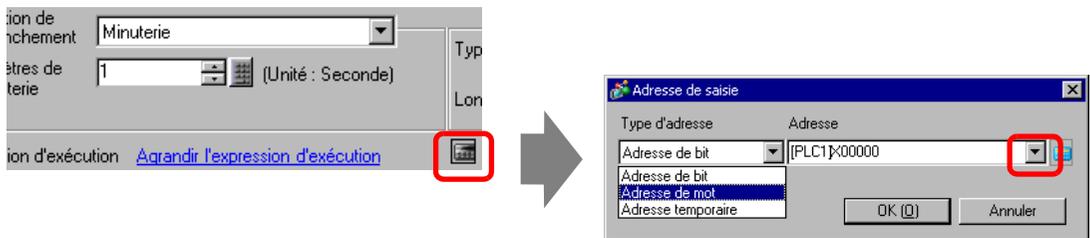
4 Dans [Déclenchement], sélectionnez [Minuterie] et définissez les [Paramètres de minuterie] sur 1 seconde.

5 Cliquez sur l'onglet [Boîte à outils]. La boîte à outils vous permet de placer facilement une commande à utiliser dans le script.

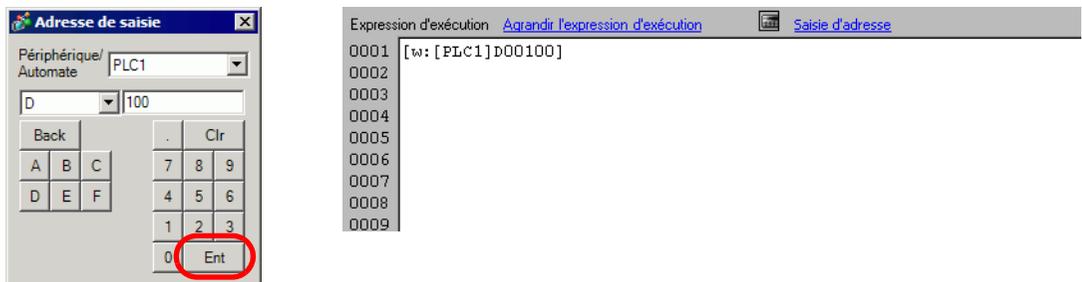


6 Créez la première ligne du script. Si vous définissez la valeur par défaut D00100 sur 0, la première opération de ligne est une opération de comptage qui s'incrémente à chaque fois qu'un processus se termine.

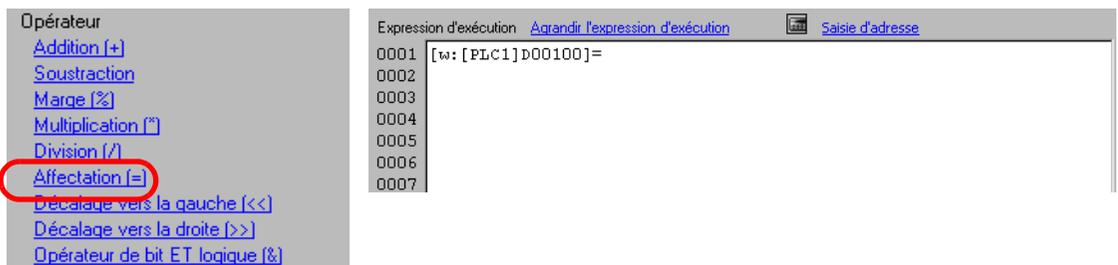
Cliquez sur  et sélectionnez [Adresse de mot], puis cliquez sur .



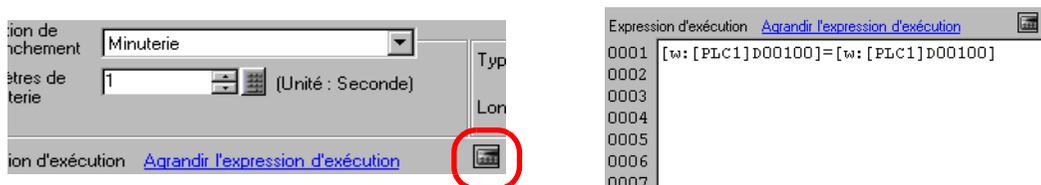
7 Saisissez D00100, puis cliquez sur [ENT]. Cliquez sur [OK] dans la boîte de dialogue [Entrée d'adresse].



8 Cliquez sur [Affectation (=)] dans la boîte à outils.



9 Placez D00100 de la même façon qu'aux étapes 6 à 7.



10 Cliquez sur [Addition (+)] et saisissez «1». La première ligne est maintenant terminée.

11 Créez la deuxième ligne du script. Dans la deuxième ligne, lorsqu'une condition est vraie, le processus qui suit l'instruction «if ()» est exécuté. Cliquez sur [if - endif].

12 Créez l'expression conditionnelle entre parenthèses «()» à la suite de «if». L'expression conditionnelle compare la valeur stockée dans D00100 à la valeur «3», et s'exécute si la condition est vraie. Placez le curseur à l'intérieur des parenthèses «()» et répétez les étapes 6 et 7 pour placer un autre D00100.

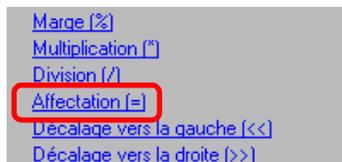
13 Cliquez sur [Equivalent (==)], puis saisissez «3». La deuxième ligne est maintenant terminée.

14 Placez le curseur à l'intérieur des parenthèses « { } » et appuyez sur la touche ENT. Répétez les étapes 6 à 7 pour placer un LS0008 supplémentaire.

```

Expression d'exécution  Agrandir l'expression d'exécution
0001 [w:[PLC1]D00100]=[w:[PLC1]D00100]+1
0002 if ([w:[PLC1]D00100]==3)
0003 {
0004     [w:[#INTERNAL]LS0008]
0005 }
0006 endif
0007
  
```

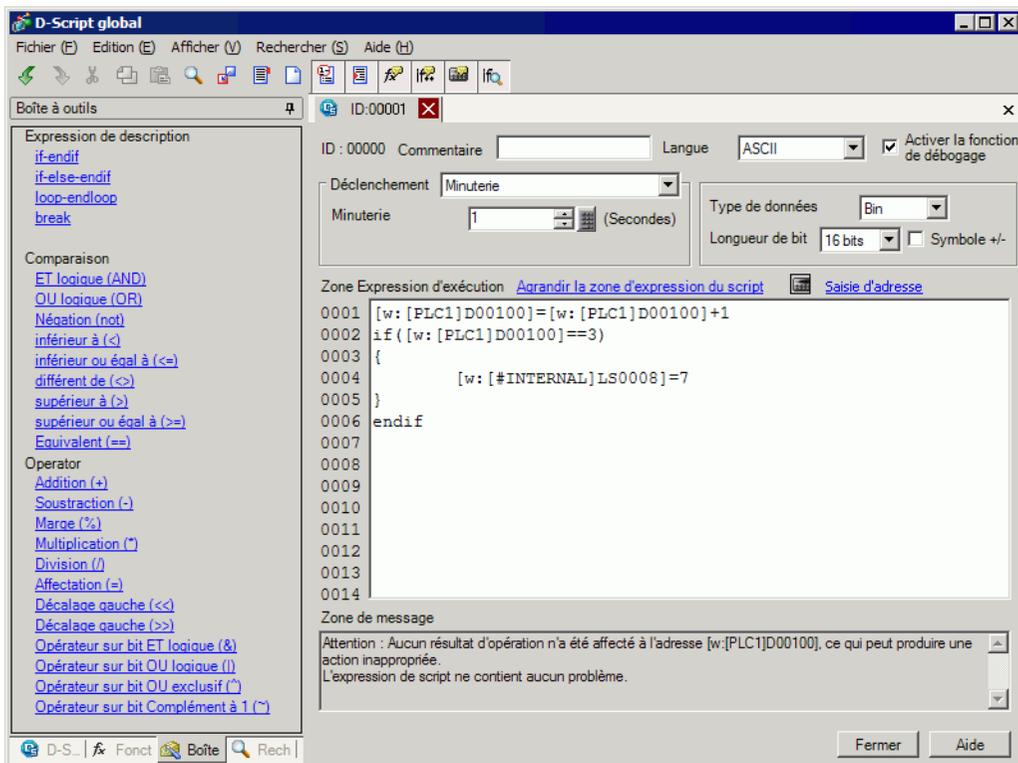
15 Cliquez sur [Affectation (=)], puis saisissez «7».



```

Expression d'exécution  Agrandir l'expression d'exécution  Saisie d'adresse
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 Le script est maintenant terminé.



REMARQUE

- Lorsque vous sélectionnez du texte, appuyez sur les touches [Ctrl] + [Maj] + la touche de flèche droite ou de flèche gauche pour sélectionner un bloc de texte complet.
- Appuyez sur les touches [Ctrl] + [F4] pour fermer l'affichage actuellement sélectionné.
- Appuyez sur la touche [Ech] pour enregistrer ou non le script, puis le fermer.

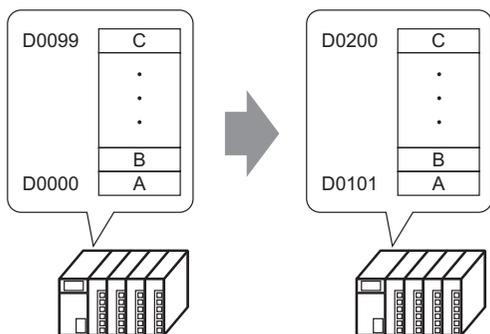
21.3 Copie des données en blocs

REMARQUE

- Pour en savoir plus, reportez-vous au guide de configuration.
 - ☞ «21.9.1 Guide de configuration du D-Script/Outils avancés [D-Script global]» (page 21-54)
- Reportez-vous à ce qui suit pour obtenir plus d'informations sur les commandes disponibles pour les scripts.
 - ☞ «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69)

Action

Créez un script qui détecte le front montant (de 0 à 1) de l'adresse de bit M0100 et copie les données stockées dans le périphérique connecté dans une autre adresse.

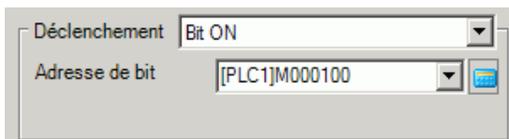


Commandes utilisées

| Commande | Résumé de la fonction |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Copie de mémoire memcpy () | <p>Copie une valeur stockée dans un périphérique dans une opération. Les données pour le nombre d'adresses seront copiées vers les adresses de mot cible à compter de la première adresse de mot des données sources.</p> <p>[Format] memcpy ([Adresse Copier vers], [Adresse Copier à partir], Mots)</p> <p>☞ «21.11.3 Opération de mémoire» (page 21-79)</p> |

Déclenchement

Dans [Déclenchement], sélectionnez [Bit ON], puis définissez [M000100] comme [Adresse de bit].



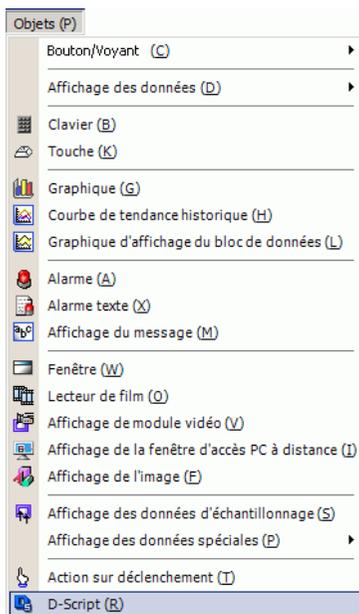
Script terminé

```

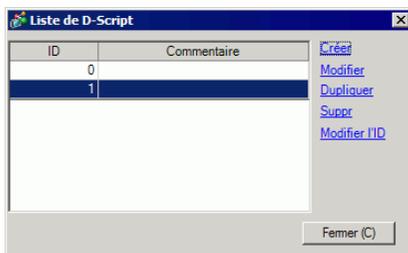
Expression d'exécution  Agrandir l'expression d'exécution  Saisie d'adresse
0001 memcpy([w:[PLC1]D00101], [w:[PLC1]D00000], 100)
0002
0003
0004
0005
0006
0007
0008
  
```

Procédure de création

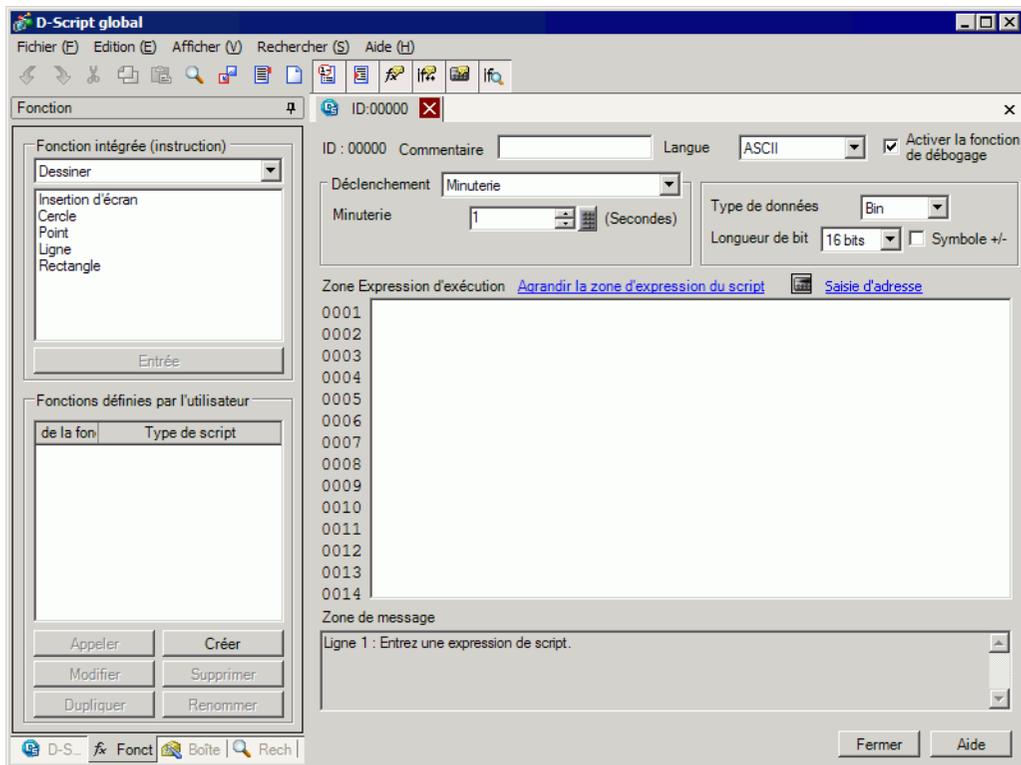
1 Dans le menu [Objets (P)], sélectionnez [D-Script (R)] ou cliquez sur  à partir de la barre d'outils.



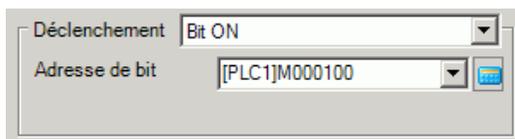
2 Cliquez sur [Créer]. Les ID des scripts existants s'affichent dans la [Liste de D-Scripts].



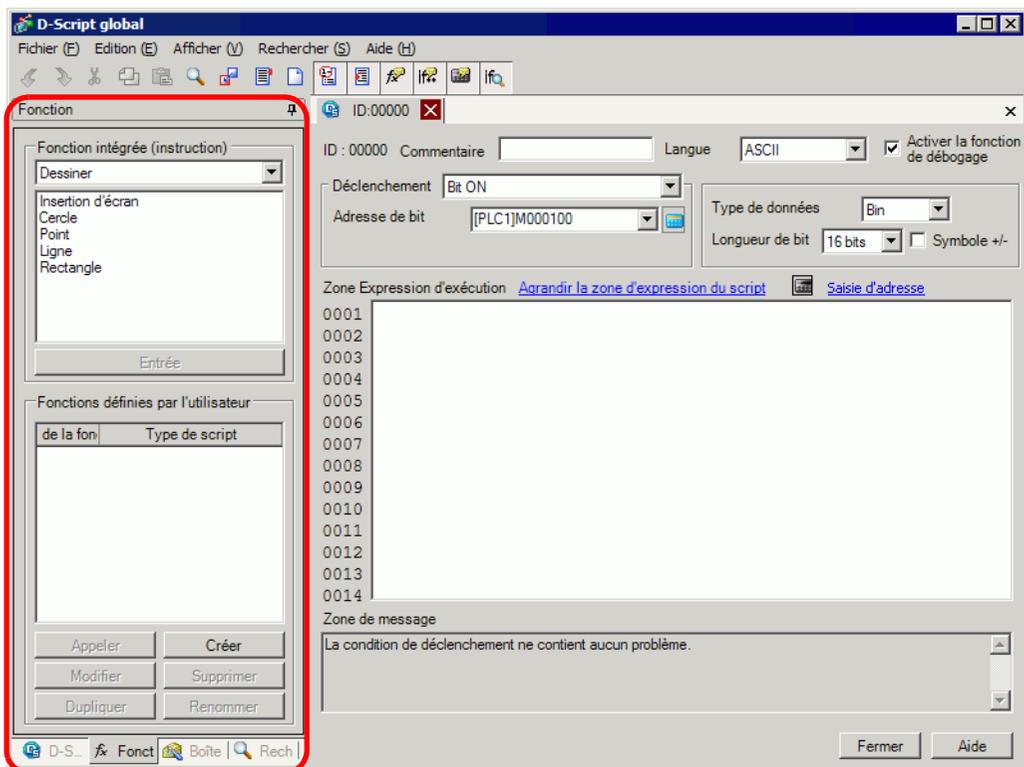
3 La boîte de dialogue [D-Script] s'affiche.



4 Sélectionnez [Bit ON] dans [Déclenchement] et précisez M000100 comme [Adresse de bit].

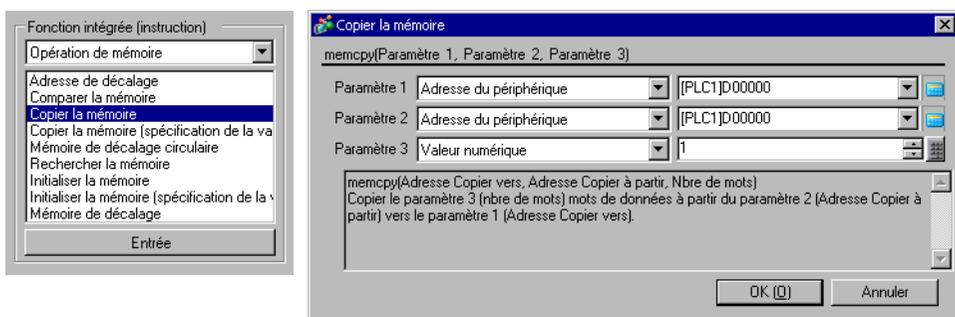


5 Cliquez sur l'onglet [Fonction]. Les fonctions intégrées vous permettent de placer facilement une commande à utiliser dans le script.

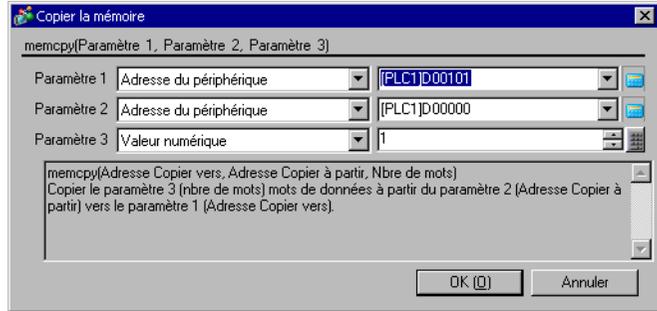
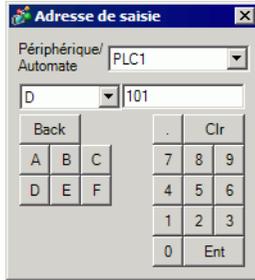


6 Dans [Fonction intégrée (Instruction)], sélectionnez [Opération mémoire].

7 Double-cliquez sur [Copie de mémoire], et dans la boîte de dialogue qui s'ouvre, définissez les paramètres pour l'adresse cible, l'adresse source et le nombre de mots. Cliquez sur .

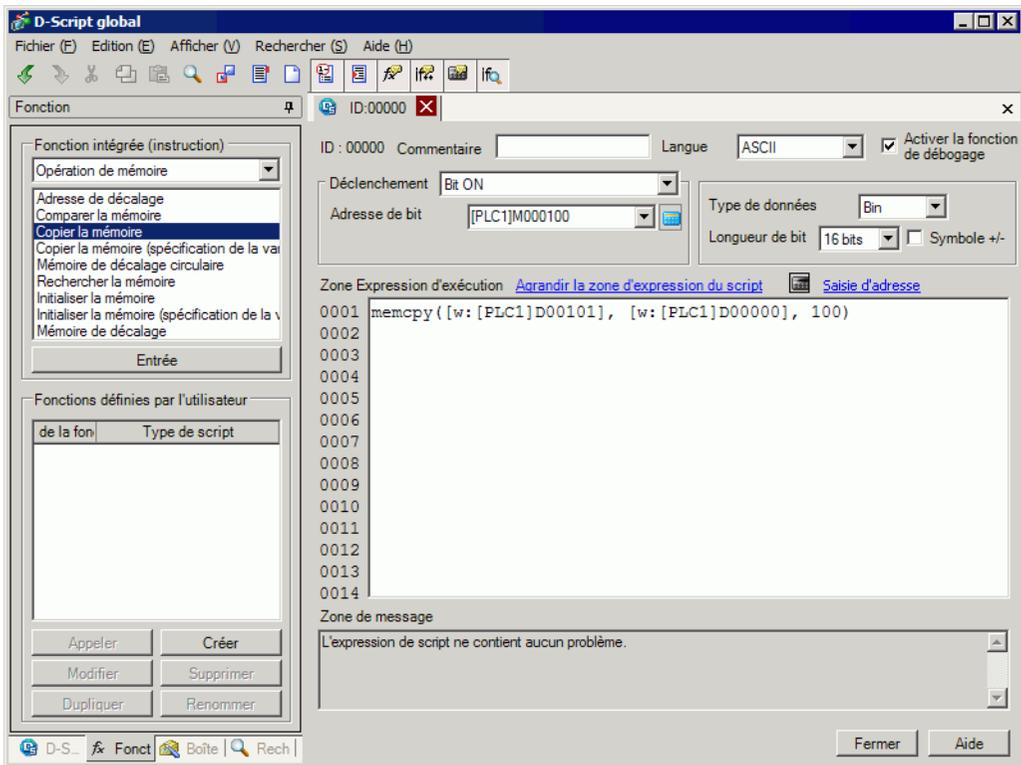


8 Pour le [Paramètre 1], saisissez D00101, et cliquez sur [ENT].



9 Pour le [Paramètre 2], entrez D00000, et cliquez sur [OK].

10 Le script est maintenant terminé.



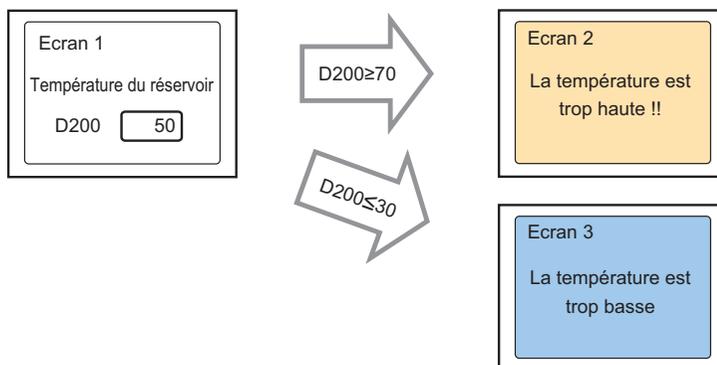
21.4 Affichage d'une alarme lorsqu'une erreur se produit

REMARQUE

- Pour en savoir plus, reportez-vous au guide de configuration.
☞ «21.9.1 Guide de configuration du D-Script/Outils avancés [D-Script global]» (page 21-54)
- Reportez-vous à ce qui suit pour obtenir plus d'informations sur les commandes disponibles pour les scripts.
☞ «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69)

Action

Le système de gestion de températures détecte un bit d'erreur dans le périphérique connecté et affiche des messages d'alarme lorsque l'adresse de stockage de l'information sur la température (D200) monte à 70° C ou plus, ou descend à 30° C ou moins. De plus, ce script compte le nombre d'erreurs détectées.



L'adresse qui compte chaque fois que D200 monte à 70° C ou plus et qui stocke le nombre de fois : LS0300

L'adresse qui compte chaque fois que D200 descend à 30° C ou moins et qui stocke le nombre de fois : LS0301

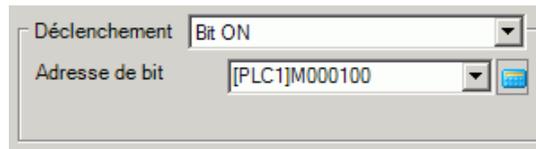
Adresse qui stocke le numéro d'écran de l'alarme : LS0008

Commandes utilisées

| Commande | Résumé de la fonction |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| if () | Si la condition «if» entre parenthèses «()», est vraie, l'expression à la suite de l'instruction «if ()» est exécutée. ☞ «21.11.8 Expressions conditionnelles» (page 21-149) |
| Supérieur ou égal à (>=) | Vrai si N1 est supérieur ou égal à N2 (N1 >= N2). ☞ «21.11.9 Comparaison» (page 21-154) |
| Affectation (=) | Affecte la valeur du côté droit dans le côté gauche. ☞ «21.11.10 Opérateur» (page 21-156) |
| Addition (+) | Ajoute une constante aux données du périphérique de mot. ☞ «21.11.10 Opérateur» (page 21-156) |
| Inférieur ou égal à (<=) | Vrai si N1 est inférieur ou égal à N2 (N1 <= N2). ☞ «21.11.9 Comparaison» (page 21-154) |

Déclenchement

Dans [Déclenchement], sélectionnez [Bit ON], puis définissez [M000100] comme [Adresse de bit].



Script terminé

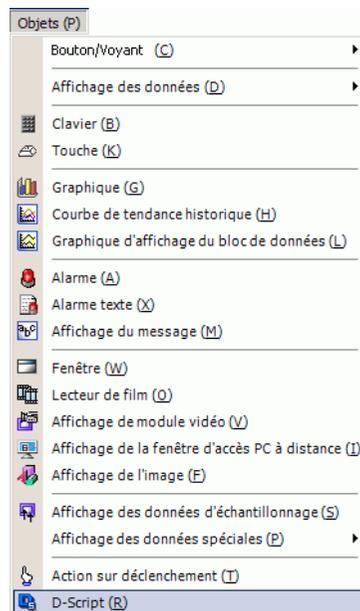
```

Expression d'exécution  Agrandir l'expression d'exécution    Saisie d'adresse
0001  if ([w:[PLC1]D00200]>=70) //When temp is greater than 70 degrees
0002  {
0003      [w:[#INTERNAL]LS0302]=100 //Greater than 70 degrees alarm screen number 100
0004      [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1 //Increase error count
0005  }
0006  endif
0007
0008  if ([w:[PLC1]D00200]>=30) //When temp is greater than 30 degrees
0009  {
0010      [w:[#INTERNAL]LS0302]=101 //Greater than 30 degrees alarm screen number 101
0011      [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1 //Increase error count
0012  }
0013  endif
0014
0015
0016

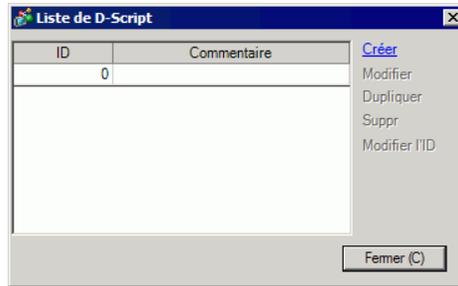
```

Procédure de création

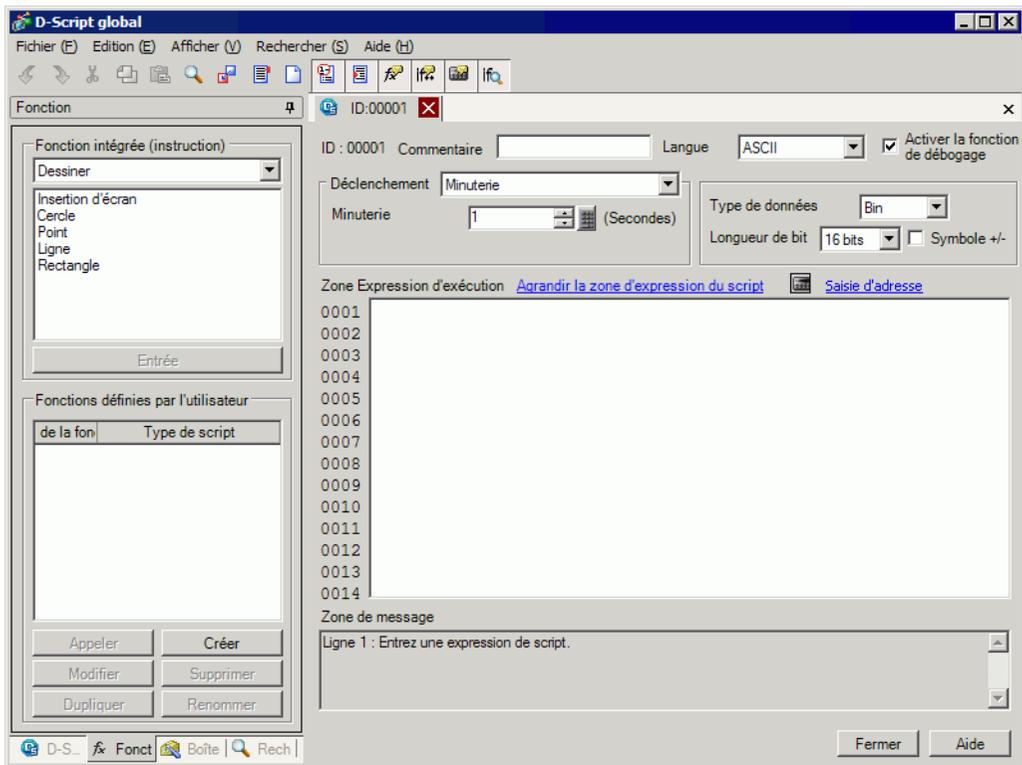
- 1 Dans le menu [Objets], cliquez sur [D-Script (R)] ou sur  .



2 Cliquez sur [Créer]. Les ID des scripts existants s'affichent dans la [Liste de D-Scripts].



3 La boîte de dialogue [D-Script] s'affiche.



4 Configurez les commentaires. Entrez «Affichage d'alarme».

5 Dans [Déclenchement], sélectionnez [Bit ON], et précisez M00100 comme [Adresse de bit].

6 Créez un programme en ajoutant des fonctions, des instructions et des expressions dans la zone d'expression de script pour compléter le script.

```
Expression d'exécution Agrandir l'expression d'exécution  Saisie d'adresse
0001 if ([w:[PLC1]D00200]>=70) //When temp is greater than 70 degrees
0002 {
0003     [w:[#INTERNAL]LS0302]=100 //Greater than 70 degrees alarm screen number 100
0004     [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1 //Increase error count
0005 }
0006 endif
0007
0008 if ([w:[PLC1]D00200]>=30) //When temp is greater than 30 degrees
0009 {
0010     [w:[#INTERNAL]LS0302]=101 //Greater than 30 degrees alarm screen number 101
0011     [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1 //Increase error count
0012 }
0013 endif
0014
0015
0016
0017
```

REMARQUE

- Lorsque vous sélectionnez du texte, appuyez sur les touches [Ctrl] + [Maj] + la touche de flèche droite ou de flèche gauche pour sélectionner un bloc de texte complet.
- Appuyez sur les touches [Ctrl] + [F4] pour fermer l'affichage actuellement sélectionné.
- Appuyez sur la touche [Ech] pour enregistrer ou non le script, puis le fermer.

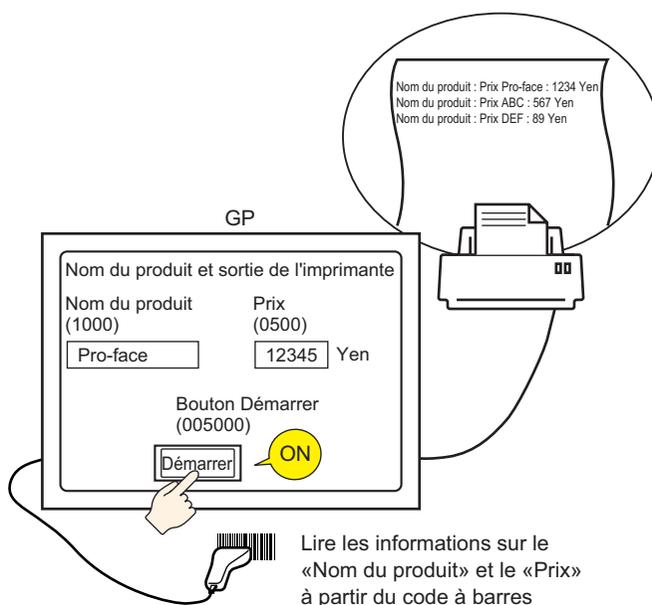
21.5 Communication avec des périphériques non pris en charge

REMARQUE

- Pour en savoir plus, reportez-vous au guide de configuration.
☞ «21.9.1 Guide de configuration du D-Script/Outils avancés [D-Script global]» (page 21-54)
- Reportez-vous à ce qui suit pour obtenir plus d'informations sur les commandes disponibles pour les scripts.
☞ «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69)

■ Opération

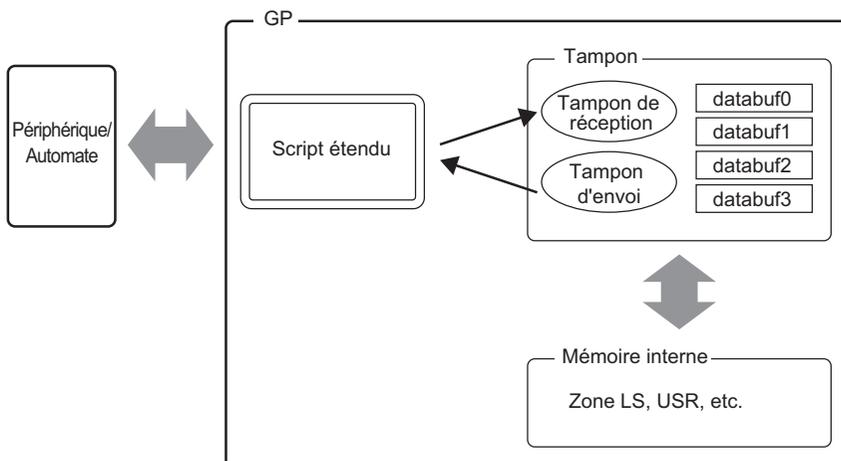
Créez un script étendu pour lire des données depuis un lecteur de code barres connecté au port USB et écrire des données vers une imprimante série connectée au port COM1.



■ Structure des scripts détaillés

Les scripts étendus sont des scripts utilisés pour établir une communication entre le port série interne du GP et les périphériques E/S connectés.

Pour la gestion de données des scripts étendus, telle qu'illustrée dans l'image suivante, les données sont stockées dans databuf0 à databuf3 via le tampon d'envoi/de réception. databuf n'est pas divisé par l'adresse, donc stockez les données dans la mémoire interne avant de modifier les données dans le périphérique/automate.



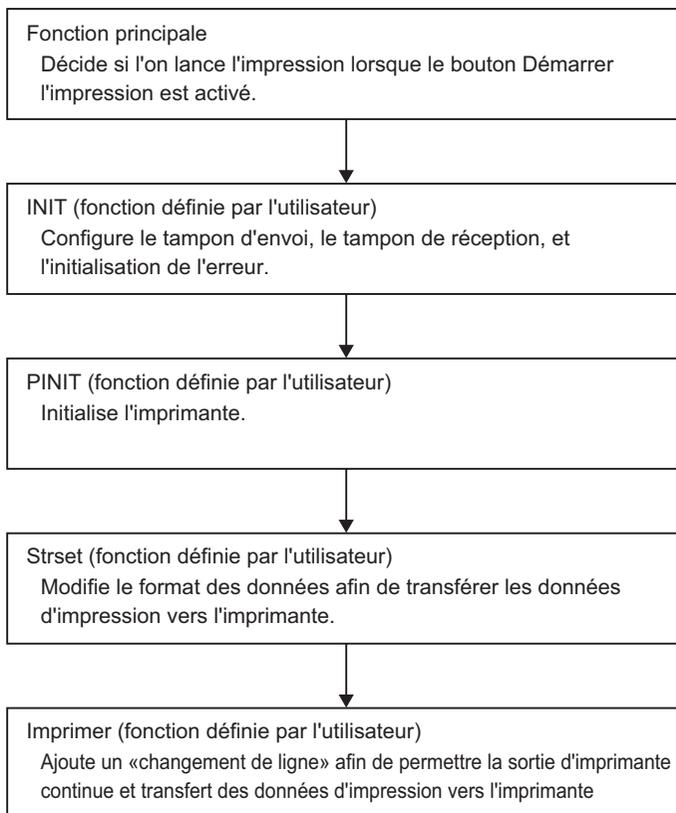
Tampon d'envoi ou de réception

Pour communiquer avec le périphérique/l'automate, cela agit en tant qu'espace de mémoire de bit qui fait une distinction entre les données reçues et envoyées en temps réel.

databuf0 à databuf3

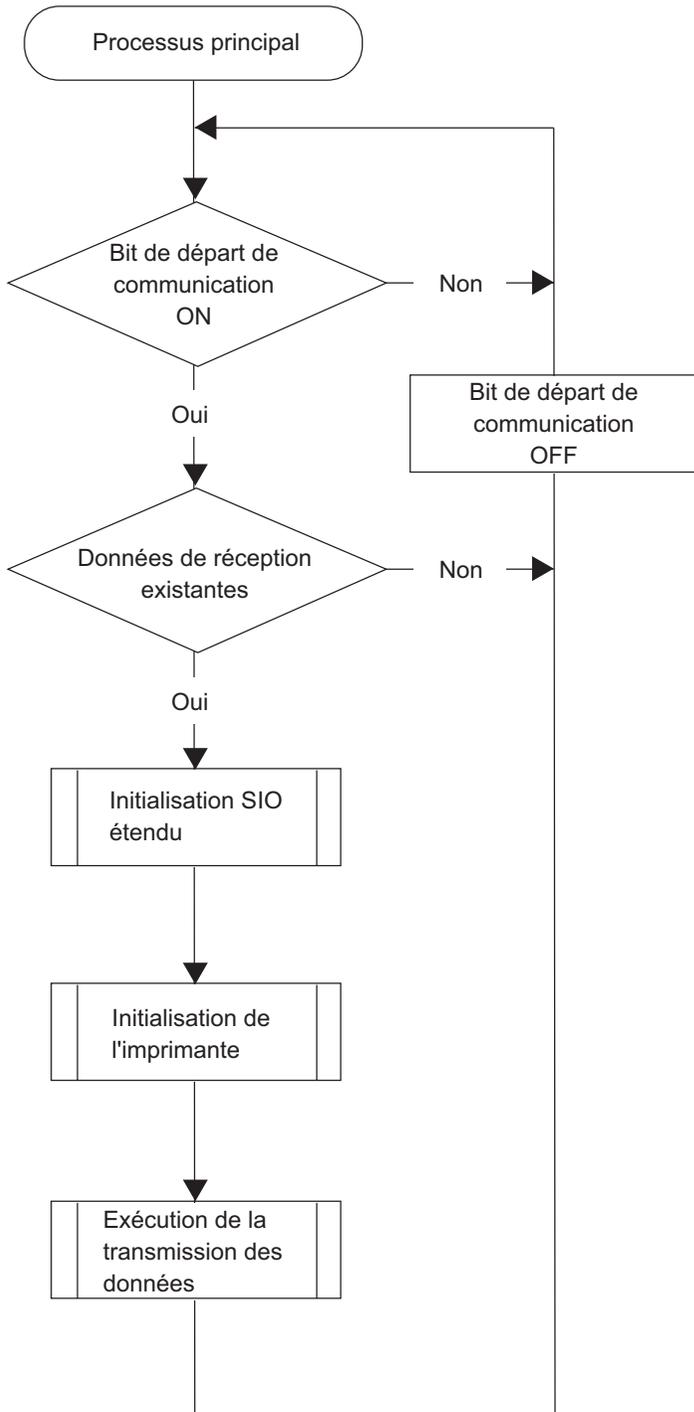
Il s'agit des espaces de mémoire d'octet (8 bits) utilisés pour le stockage des données. La taille du tampon est de 1 Ko.

■ Procédure de création des scripts

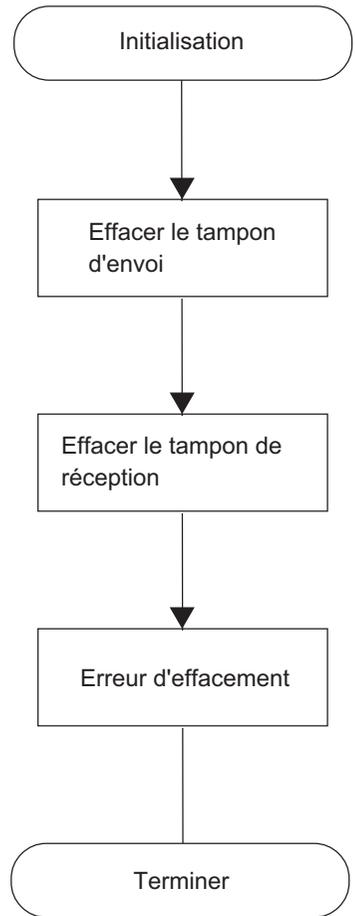


■ Organigramme

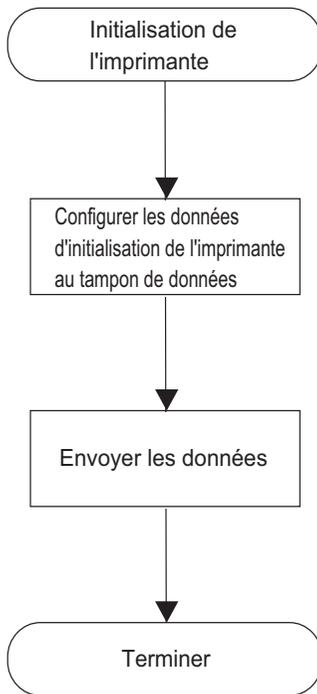
(1) Processus principal



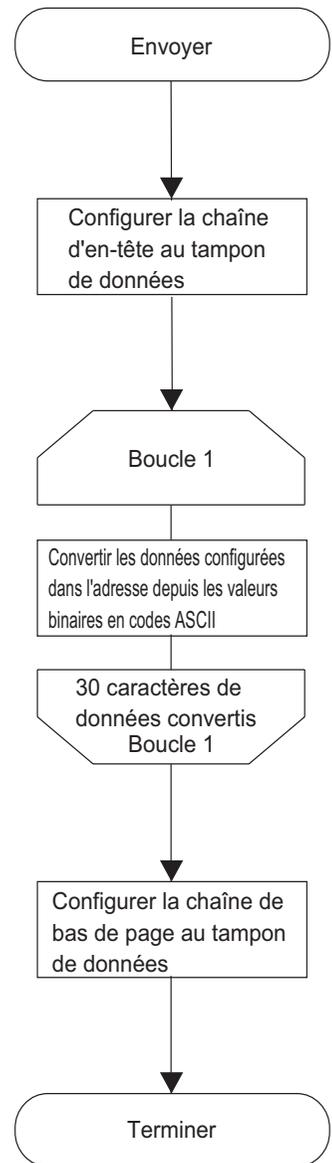
(2) Fonction d'initialisation (INT)



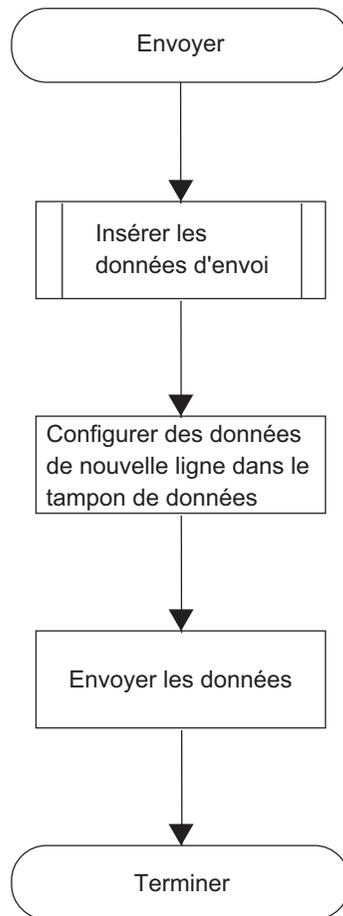
(3) Fonction d'initialisation de l'imprimante (PINIT)



(4) Fonction de chaîne (Strset)



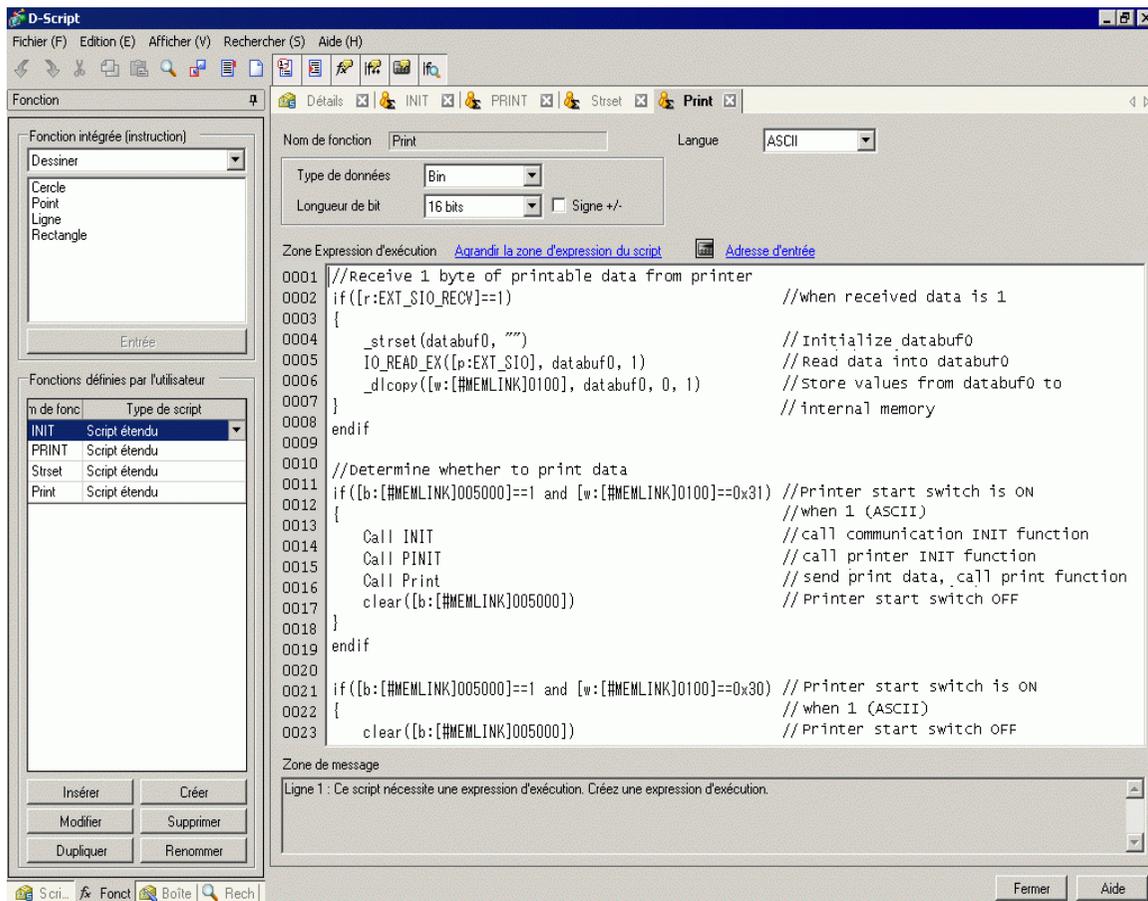
(5) Fonction d'envoi (imprimer)



■ Aperçu du fonctionnement des scripts

◆ Fonctions principales

Script terminé



Résumé de la fonction

Lorsque le bouton de démarrage de l'impression (mémoire interne 005000) est activé, le script décide s'il faut démarrer l'impression à compter du premier octet de données autorisées d'impression.

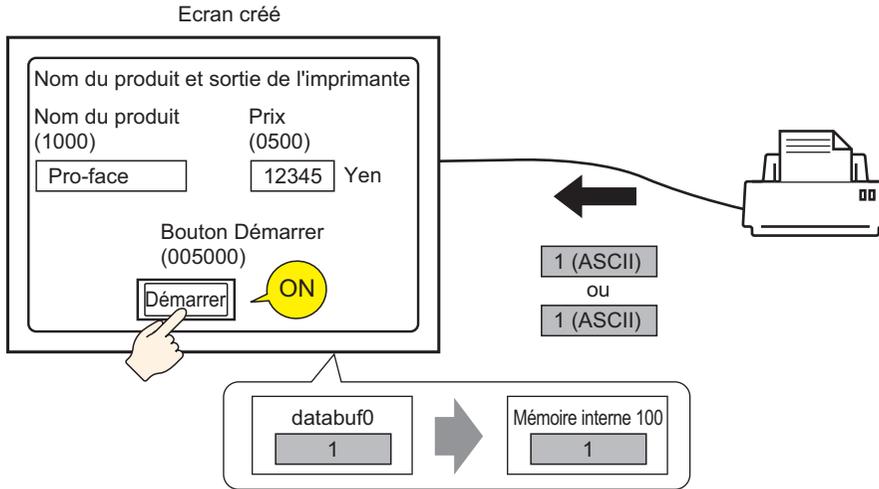
Les données d'impression autorisée effectuent les actions suivantes comme exemple de spécifications d'imprimante.

Préparation d'impression OK : Envoyez 0x31 (code ASCII «1») vers le périphérique/l'automate.

Préparation d'impression non valide : Envoyez 0x30 (code ASCII «0») vers le périphérique/l'automate.

Le GP reçoit les données d'impression autorisée dans databuf0 et ces données sont déplacées vers la mémoire interne 100 à l'aide du traitement de script suivant.

Si la mémoire interne 100 = 0x31 (code ASCII de «1»), l'impression démarre. Si la mémoire interne est 0x30 (code ASCII de «0»), le GP relance le script et répète ce processus jusqu'à ce que les données 0x31 soient reçues.



◆ INIT (fonction définie par l'utilisateur)

Script terminé

```

Expression d'exécution  Agrandir l'expression d'exécution  Saisie d'adresse
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
    
```

Résumé de la fonction

Configure le tampon d'envoi, le tampon de réception, et l'initialisation de l'erreur.

◆ PINIT (fonction définie par l'utilisateur)

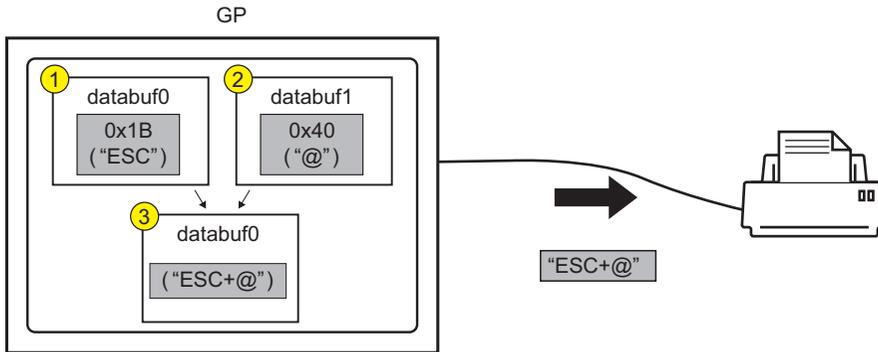
Script terminé

```

Zone Expression d'exécution  Agrandir la zone d'expression du script  Adresse d'entrée
0001 Call Strset //call Printing Data function
0002 _strset(databuf0,"") //clear databuf0
0003
0004 //Printing and Delimiter (Carriage Return / Line Feed)
0005
0006 _strset(databuf0,0x0d) //Print out ,then go back to first place of the line
0007 _strcat(databuf1,databuf0) //append databuf0 into the end of databuf1
0008 _strset(databuf0,"") //clear databuf0
0009 _strset(databuf0,0x0a) //go to next line
0010 _strcat(databuf1,databuf0) //append databuf0 into the end of databuf1
0011
0012 _strlen([t:0000],databuf1) //store data length to temporary address
0013
0014 //Send data over serial port
0015
0016 IO_WRITE_EX([p:EXT_SIO],databuf1,[t:0000]) //Send databuf0, amount defined by temporary address value
0017
    
```

Résumé de la fonction

Initialise l'imprimante. Envoyez la commande ESC/P «ESC+@» vers l'imprimante.



◆ Strset (fonction définie par l'utilisateur)

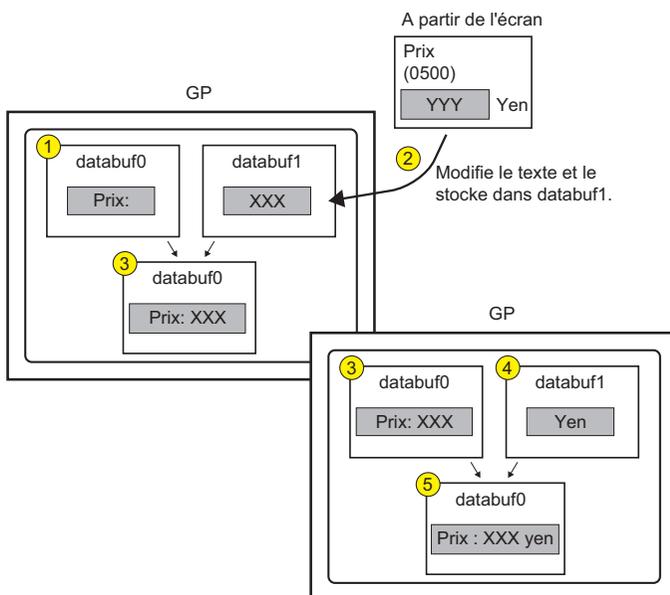
Script terminé

```

Zone Expression d'exécution  Agrandir la zone d'expression du script  Adresse d'entrée
0001 //String example, add "Price:" and "$"
0002 _strset(databuf0, "") //Initialize databuf0
0003 _strset(databuf0, "Price:") //Store text "Price:" to databuf0
0004 _bin2decase(databuf0, [w:[#MEMLINK] 0500]) //Convert value to string and store in databuf1
0005 _strcat(databuf0, databuf1) //Add databuf1 to end of databuf0
0006 _strset(databuf1, "") //Initialize databuf1
0007 _strset(databuf1, "$") //Store text "$" to databuf1
0008 _strcat(databuf0, databuf1) //Add databuf1 to end of databuf0
0009
0010 //Initialize temporary address
0011 [t:0001]=0
0012 [t:0002]=0
0013
0014 //Store to internal memory word units, consecutive characters into byte units(30 characters)
0015 loop()
0016 {
0017 [w:[#MEMLINK] 2000]#[t:0002]=[w:[#MEMLINK] 1000]#[t:0001]>>8 //Store top byte into bottom byte
0018 [w:[#MEMLINK] 2001]#[t:0002]=[w:[#MEMLINK] 1000]#[t:0001]& 0xFF //Erase top byte and store in next address
0019 [t:0001]=[t:0001]+1 //Address offset + 1
0020 [t:0002]=[t:0002]+2 //Address offset + 2
0021 if([t:0001]==15) //Store 2 words into 2 byte and repeat 15 times
0022 {
0023 break
0024 }
0025 endif
0026 }
0027 endloop
0028 _ldcopy(databuf2, [w:[#MEMLINK] 2000],30) //Store internal memory 2000-2030 to data buffer as characters
0029
0030 //Add string "Item:"
0031 _strset(databuf1, "") //Initialize databuf1
0032 _strset(databuf1, "Item:") //Store string "Item:" into databuf1
0033 _strcat(databuf1, databuf2) //Add databuf1 to end of databuf0
0034
0035 //Add Item and Price strings
0036 _strcat(databuf1, databuf0) //Add databuf0 to end of databuf1
    
```

Résumé de la fonction

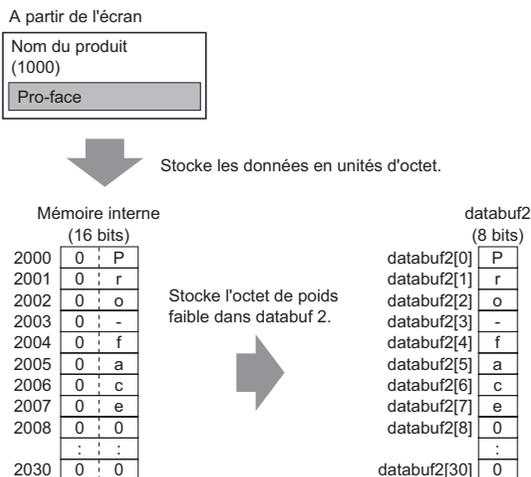
1 Ajoutez le texte «Prix :» et «Yen» aux données de prix stockées dans la mémoire interne 0500.



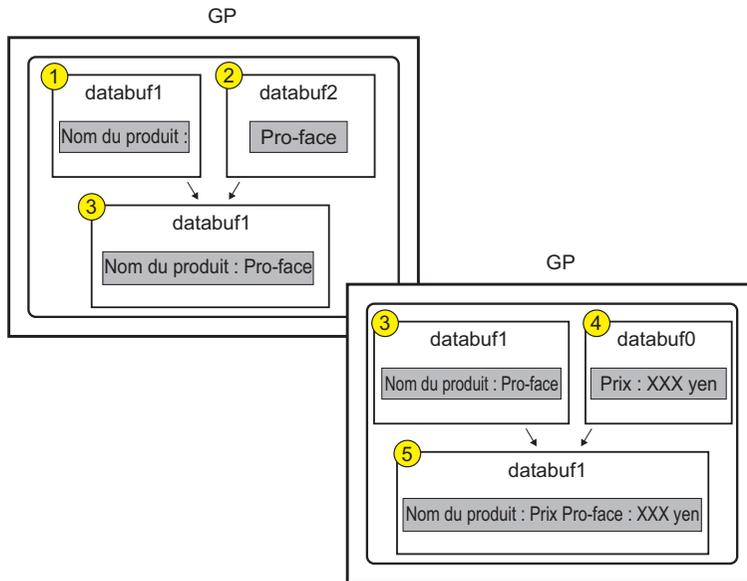
2 Modifiez le format des données afin d'envoyer les données d'impression vers l'imprimante. Divisez les données de chaîne (nom de produit) stockées de façon séquentielle dans la mémoire interne 1000 en unités d'octet, et stockez-les dans les mémoires internes 2000 à 2030 en tant que données de chaîne de poids faible. Utilisez la fonction `_ldcopy` et stockez l'octet de poids faible dans `databuff2`.

REMARQUE

- La fonction `_ldcopy` prend les données stockées en tant que mots, et ne stocke que les octets de poids faible dans le tampon, tandis que les données d'octet de poids fort sont ignorées.



3 Ajoutez le texte «Nom de produit :» et «Prix» au databuf2.



◆ Imprimer (fonction définie par l'utilisateur)

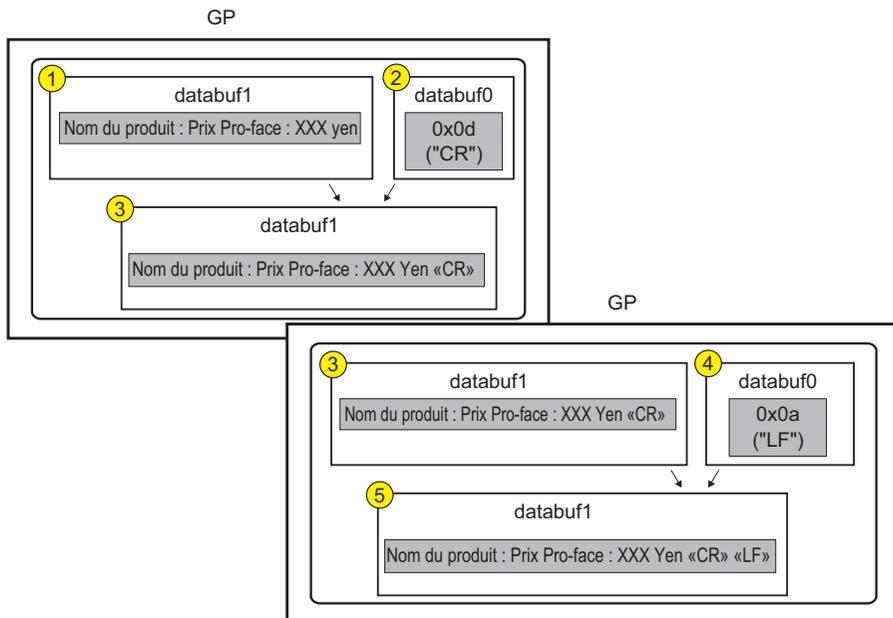
Script terminé

```

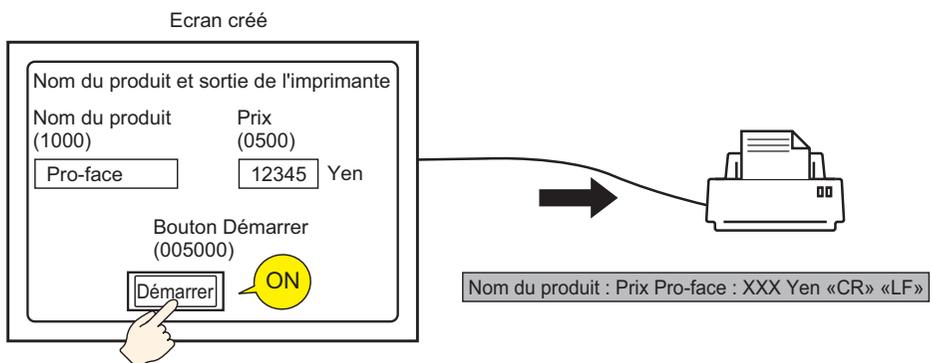
Zone Expression d'exécution  Agrandir la zone d'expression du script  Adresse d'entrée
0001 Call Street //Call string data function
0002 _strset(databuf0,"") //Clear databuf1
0003
0004 //Text delimiter
0005
0006 _strset(databuf0, 0*0d) //Return to start of row
0007 _strset(databuf1, databuf0) //Add databuf1 to end of databuf0
0008 _strset(databuf0, "") //Clear databuf1
0009 _strset(databuf0, 0*0a) //New line
0010 _strset(databuf1, databuf0) //Add databuf1 to end of databuf0
0011
0012 _strset([t:0000], databuf1) //Store data length to temporary address
0013
0014 //Send data over serial port
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000] //Send databuf0, amount defined by temporary address value
0017
    
```

Résumé de la fonction

1 Ajoutez un «changement de ligne» afin de permettre une impression continue.



2 Envoyez les données d'impression vers l'imprimante.

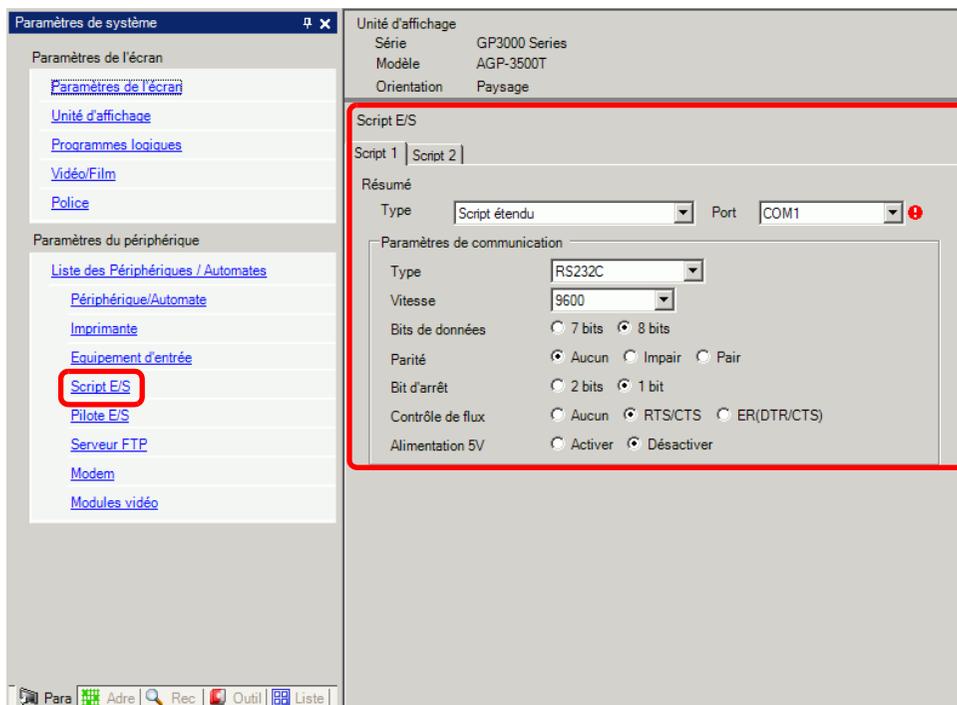


■ Commandes utilisées

| Commande | Résumé de la fonction |
|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| if () | Si la condition «if» entre parenthèses «()», est vraie, l'expression à la suite de l'instruction «if ()» est exécutée. ☞ «21.11.8 Expressions conditionnelles» (page 21-149) |
| Paramètres d'étiquette [r:EXT_SIO_RECV] | Indique la quantité des données (nombre d'octets) reçues à ce moment. La taille des données reçues est en lecture seule. ☞ «21.11.4 Opération de port SIO» (page 21-100) |
| Equivalent (==) | Vrai si N1 est égal à N2 (N1 = N2). ☞ «21.11.9 Comparaison» (page 21-154) |
| Paramètres de texte (_strset) | Une chaîne fixe est stockée dans le tampon de données. ☞ «21.11.11 Opération de texte» (page 21-160) |
| Réception étendue (IO_READ_EX) | Reçoit les données de la taille indiquée dans Taille des données reçues (octets) depuis SIO étendu et les stocke dans le tampon de données. ☞ «21.11.4 Opération de port SIO» (page 21-100) |
| A partir du tampon de données vers la variable interne (_dlcopy) | Chaque octet de données de chaîne stocké dans le décalage du tampon de données est copié vers la zone LS selon le nombre de chaînes. ☞ «21.11.11 Opération de texte» (page 21-160) |
| Paramètres d'étiquette [c:EXT_SIO_CTRL**] | Cette variable de contrôle sert à effacer le tampon d'envoi, le tampon de réception, et à contrôler le statut d'erreur. ☞ «21.11.8 Expressions conditionnelles» (page 21-149) |
| Relier le texte (_strcat) | Une chaîne de caractères ou un code caractère est concaténé avec le tampon de texte. ☞ «21.11.11 Opération de texte» (page 21-160) |
| Longueur du texte (_strlen) | Obtient la longueur de la chaîne stockée. ☞ «21.11.11 Opération de texte» (page 21-160) |
| Script étendu (IO_WRITE_EX) | Envoie les données dans le tampon de données avec le SIO étendu selon la taille du nombre d'octets d'envoi. ☞ «21.11.4 Opération de port SIO» (page 21-100) |
| Affectation (=) | Affecte la valeur du côté droit dans le côté gauche. ☞ «21.11.10 Opérateur» (page 21-156) |
| Addition (+) | Ajoute une constante aux données du périphérique de mot. ☞ «21.11.10 Opérateur» (page 21-156) |
| Conversion de chaîne décimale de la valeur numérique (_bin2decasc) | Cette fonction sert à convertir un entier en une chaîne décimale. ☞ «21.11.11 Opération de texte» (page 21-160) |
| A partir de la variable interne vers le tampon de données (_ldcopy) | Les données de la chaîne stockées dans la zone LS sont copiées vers le tampon de données selon le nombre de chaînes dans un transfert octet par octet. ☞ «21.11.11 Opération de texte» (page 21-160) |

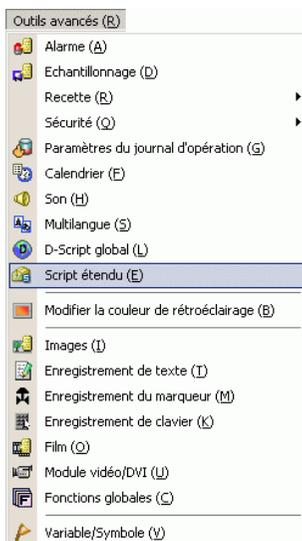
■ Procédure de création

- 1 Configurez le script pour qu'il utilise un script étendu pour la communication. Dans le menu [Projet (F)], cliquez sur [Paramètres système (C)]. Sélectionnez [Script]. Configurez le [Type] à [Script étendu].



Il y a deux onglets pour les paramètres de script. Configurez le [Port] à COM1 ou COM2. Configurez les [Paramètres de communication] afin de correspondre au SIO étendu.

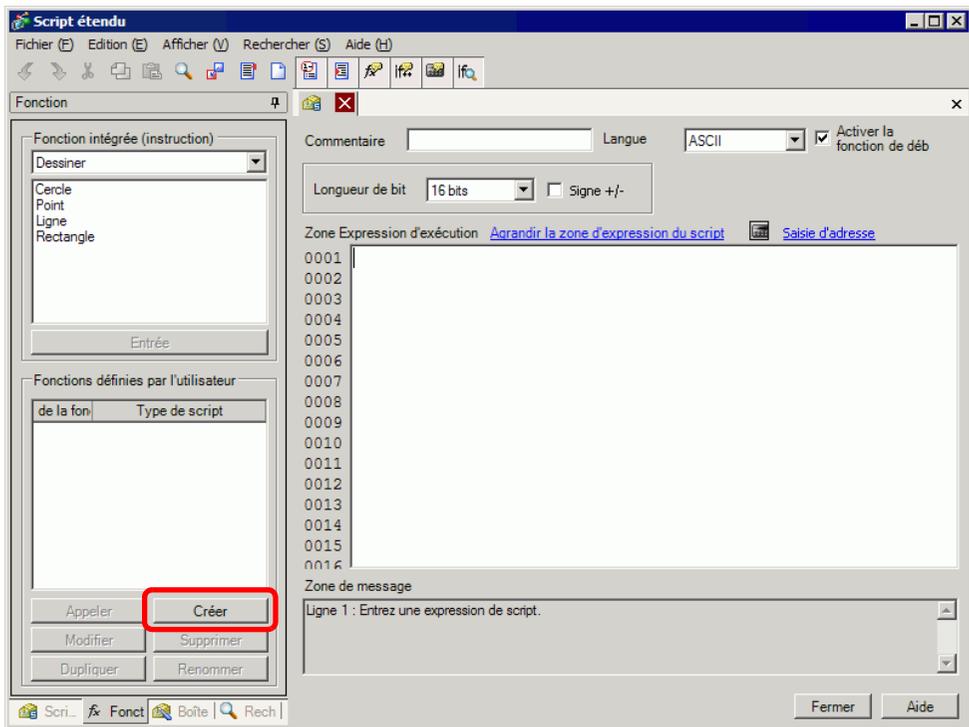
- 2 Dans le menu [Outils avancés (R)], sélectionnez [Script étendu (E)].



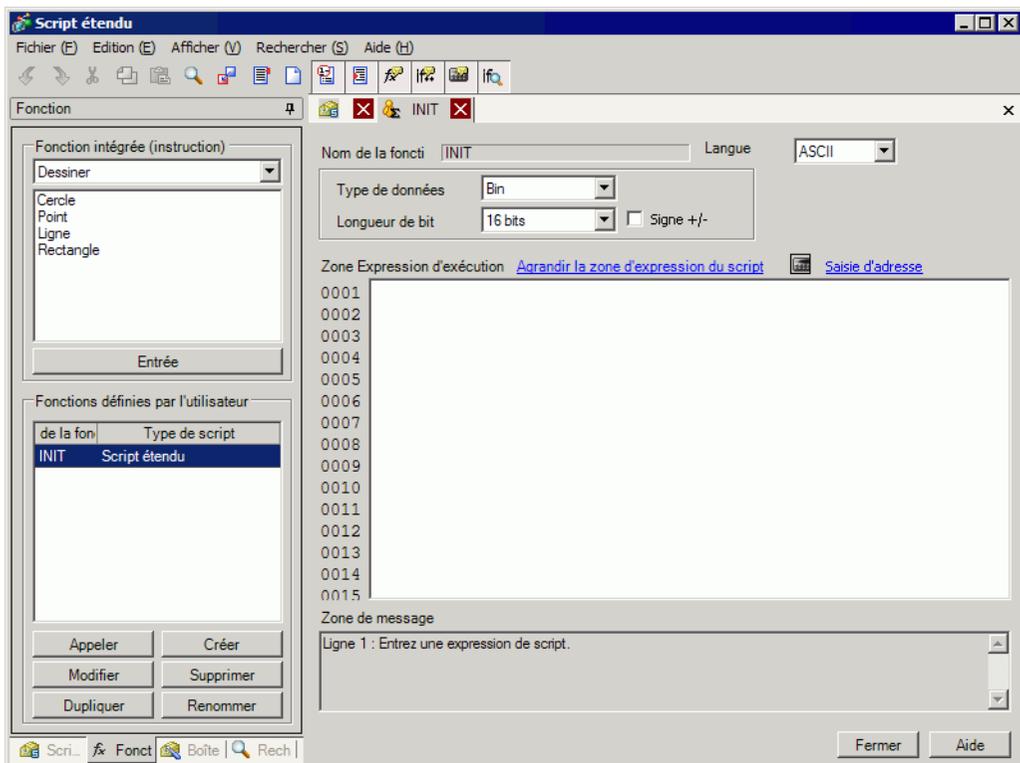
REMARQUE

- Lorsque le message «Le script étendu sera utilisé. Continuer ?» apparaît, cliquez sur [Oui].

3 Enregistrez «INIT» en tant que fonction définie par l'utilisateur. Cliquez sur l'onglet [Fonction], puis cliquez sur le bouton [Créer] du cadre de fonction définie par l'utilisateur.



4 Saisissez [INIT] en tant que nom de fonction, puis cliquez sur [OK]. L'écran suivant apparaît :



5 Créez un script dans l'expression d'exécution à l'aide de commandes, d'instructions, et la saisie continue.

```

Expression d'exécution  Agrandir l'expression d'exécution  Saisie d'adresse
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 De la même manière, enregistrez «PINIT» en tant que fonction définie par l'utilisateur. Saisissez [PINIT] en tant que nom de fonction, puis créez le script suivant dans l'expression d'exécution.

```

Zone Expression d'exécution  Agrandir la zone d'expression du script  Adresse d'entrée
0001 Call Strset //call Printing Data function
0002 _strset(databuf0,"") //clear databuf0
0003
0004 //Printing and Delimiter (Carriage Return / Line Feed)
0005
0006 _strset(databuf0,0x0d) //Print out ,then go back to first place of the line
0007 _strcat (databuf1,databuf0) //append databuf0 into the end of databuf1
0008 _strset (databuf0,"") //clear databuf0
0009 _strset (databuf0,0x0a) //go to next line
0010 _strcat (databuf1,databuf0) //append databuf0 into the end of databuf1
0011
0012 _strlen([t:0000],databuf1) //store data length to temporary address
0013
0014 //Send data over serial port
0015
0016 IO_WRITE_EX([p:EXT_SIO],databuf1,[t:0000]) //Send databuf0, amount defined by temporary address value
0017
    
```

7 De la même manière, enregistrez «Strset» en tant que fonction définie par l'utilisateur. Saisissez [Strset] en tant que nom de fonction, puis créez le script suivant dans l'expression d'exécution.

```

Zone Expression d'exécution  Agrandir la zone d'expression du script  Adresse d'entrée
0001 //String example, add "Price:" and "$"
0002 _strset (databuf0, "") //Initialize databuf0
0003 _strset (databuf0, "Price:") //Store text "Price:" to databuf0
0004 _bin2decasc (databuf0,[w:[#MEMLINK]0500]) //Convert value to string and store in databuf1
0005 _strcat (databuf0, databuf1) //Add databuf1 to end of databuf0
0006 _strset (databuf1, "") //Initialize databuf1
0007 _strset (databuf1, "$") //Store text "$" to databuf1
0008 _strcat (databuf0, databuf1) //Add databuf1 to end of databuf0
0009
0010 //Initialize temporary address
0011 [t:0001]=0
0012 [t:0002]=0
0013
0014 //Store to internal memory word units, consecutive characters into byte units(30 characters)
0015 loop()
0016 {
0017 [w:[#MEMLINK]2000]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001]>>8 //Store top byte into bottom byte
0018 [w:[#MEMLINK]2001]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001]& 0xFF //Erase top byte and store in next address
0019 [t:0001]=[t:0001]+1 //Address offset + 1
0020 [t:0002]=[t:0002]+2 //Address offset + 2
0021 if([t:0001]==15) //Store 2 words into 2 byte and repeat 15 times
0022 {
0023 break
0024 }
0025 endif
0026 }
0027 endloop
0028 _ldcopy(databuf2, [w:[#MEMLINK]2000],30) //Store internal memory 2000-2030 to data buffer as characters
0029
0030 //Add string "Item:"
0031 _strset (databuf1, "") //Initialize databuf1
0032 _strset (databuf1, "Item:") //Store string "Item:" into databuf1
0033 _strcat (databuf1, databuf2) //Add databuf1 to end of databuf0
0034
0035 //Add Item and Price strings
0036 _strcat (databuf1, databuf0) //Add databuf0 to end of databuf1
    
```

- 8 De la même manière, enregistrez «Imprimer» en tant que fonction définie par l'utilisateur. Saisissez [PINIT] en tant que nom de fonction, puis créez le script suivant dans l'expression d'exécution.

```

Zone Expression d'exécution  Agrandir la zone d'expression du script  Adresse d'entrée
0001 Call Street //Call string data function
0002 _strset(databuf0,"") //Clear databuf1
0003
0004 //Text delimiter
0005
0006 _strset(databuf0, 0*0d) //Return to start of row
0007 _strset(databuf1, databuf0) //Add databuf1 to end of databuf0
0008 _strset(databuf0, "") //Clear databuf1
0009 _strset(databuf0, 0*0a) //New line
0010 _strset(databuf1, databuf0) //Add databuf1 to end of databuf0
0011
0012 _strset([t:0000], databuf1) //Store data length to temporary address
0013
0014 //Send data over serial port
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000] //Send databuf0, amount defined by temporary address value
0017

```

- 9 Créez le script principal. Créez le script suivant dans l'expression d'exécution pour compléter le script.

```

Zone Expression d'exécution  Agrandir la zone d'expression du script  Adresse d'entrée
0001 //Receive 1 byte of printable data from printer
0002 if([r:EXT_SIO_RECV]==1) //When received data is 1
0003 {
0004     _strset(databuf0,"") //Initialize databuf0
0005     IO_READ_EX([p:EXT_SIO], databuf0, 1) //Read data into databuf0
0006     dlcopyp([w:[#MEMLINK]0100], databuf0, 0, 1) //Store values from databuf0 to internal memory
0007 }
0008 endif
0009
0010 //Determine whether to print data
0011 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0x31) //Printer start switch is ON
0012 {
0013     Call INIT //call communication INIT function
0014     Call PINIT //call printer INIT function
0015     Call Print //send print data, call print function
0016     clear([b:[#MEMLINK]005000]) //Printer start switch OFF
0017 }
0018 endif
0019
0020 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0x30) //Printer start switch is ON
0021 {
0022     clear([b:[#MEMLINK]005000]) //Printer start switch OFF
0023 }
0024 }
0025 endif

```

REMARQUE

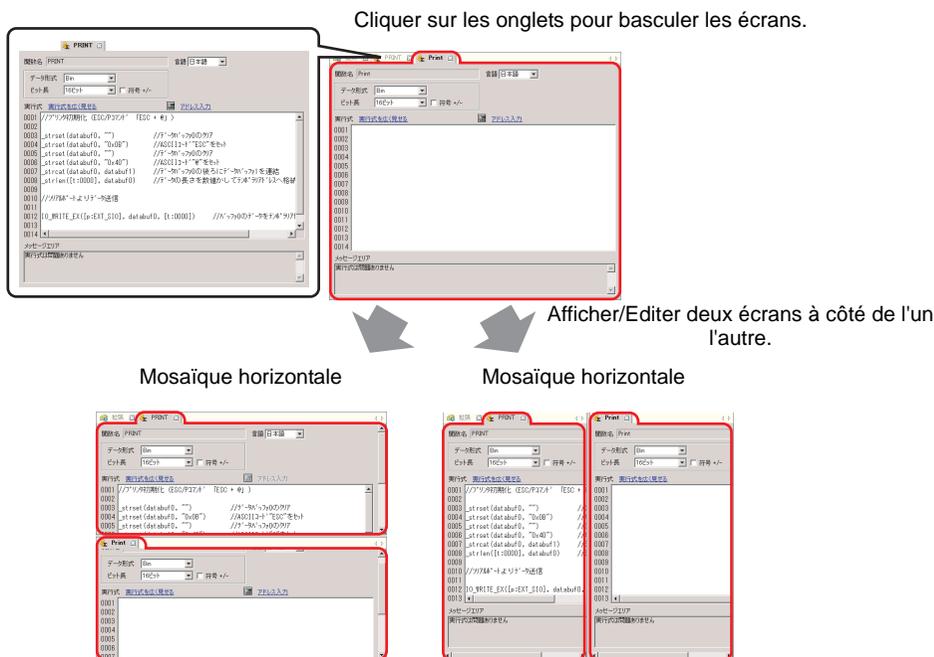
- Lorsque vous placez des fonctions définies par l'utilisateur créées dans les étapes 3 à 9 dans le script principal, sélectionnez la fonction à placer, puis cliquez sur [Insérer] dans l'onglet [Fonction]. La fonction sera placée à l'aide du «Nom de fonction d'insertion».
- Lorsque vous sélectionnez du texte, appuyez sur les touches [Ctrl] + [Maj] + la touche de flèche droite ou de flèche gauche pour sélectionner un bloc de texte complet.
- Appuyez sur les touches [Ctrl] + [F4] pour fermer l'affichage actuellement sélectionné.
- Appuyez sur la touche [Ech] pour enregistrer ou non le script, puis le fermer.

21.6 Créer une mosaïque entre deux scripts

21.6.1 Introduction

Vous pouvez afficher une fonction définie par l'utilisateur à côté d'un D-script, d'un D-script global, d'un script étendu ou d'une autre fonction définie par l'utilisateur.

Vous pouvez écrire la fonction lorsque vous les comparez, ou vous pouvez modifier les deux en même temps.



REMARQUE

- Vous pouvez également modifier des différents types de scripts définis par l'utilisateur tous en même temps.

21.6.2 Procédure d'exploitation

REMARQUE

- Pour en savoir plus, reportez-vous au guide de configuration.
 - ☞ «21.9.1 Guide de configuration du D-Script/Outils avancés [D-Script global]» (page 21-54)
- Reportez-vous à ce qui suit pour obtenir plus d'informations sur les commandes disponibles pour les scripts.
 - ☞ «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69)
- Reportez-vous à ce qui suit pour obtenir plus d'informations sur les commandes disponibles pour les scripts.
 - ☞ «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69)

Dans la boîte de dialogue [D-Script], fractionnez l'écran en deux écrans horizontaux ou verticaux.

Par exemple, lorsque la fonction créée par l'utilisateur «IMPRIMER» est affichée par une procédure précédente, créez «Imprimer».

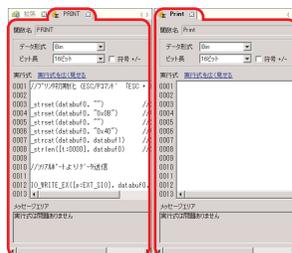
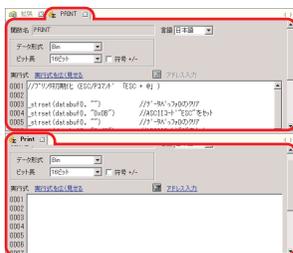
Cliquer sur les onglets pour basculer les écrans.



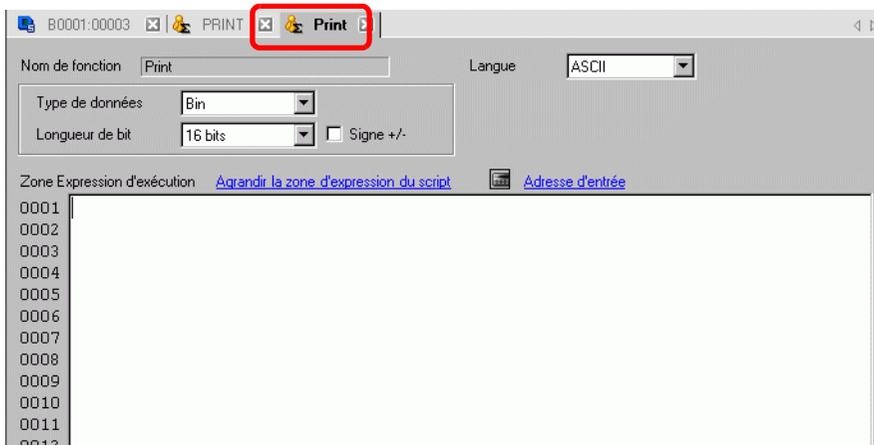
Afficher/Editer deux écrans à côté de l'un l'autre.

Mosaïque horizontale

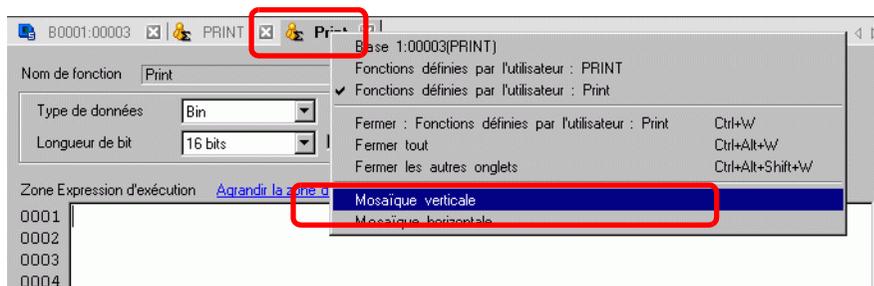
Mosaïque verticale



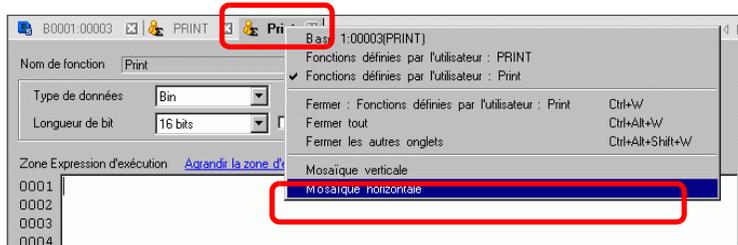
1 Dans la boîte de dialogue [D-Script], ouvrez le script et la fonction définie par l'utilisateur que vous souhaitez afficher simultanément.



2 Si vous souhaitez organiser les écrans en mosaïque horizontale, cliquez à droite sur «l'onglet d'écran à afficher en bas de la fenêtre», puis cliquez sur [Mosaïque horizontale].

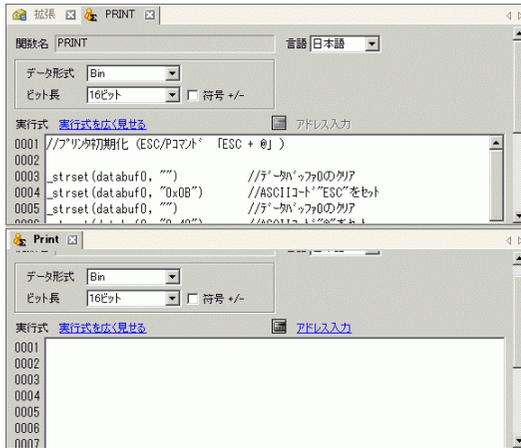


Si vous souhaitez organiser les écrans en mosaïque verticale, cliquez à droite sur «l'onglet d'écran à afficher à la droite de la fenêtre», puis cliquez sur [Mosaïque verticale].

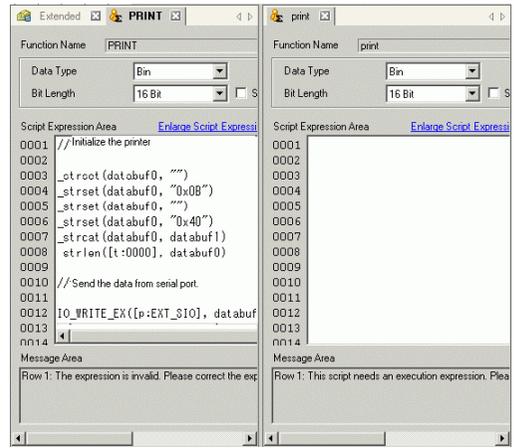


3 Les écrans sont organisés en mosaïque horizontale ou verticale.

Mosaïque horizontale



Mosaïque verticale



21.7 Création de scripts

21.7.1 Procédure pour créer des D-Scripts/D-Scripts globaux

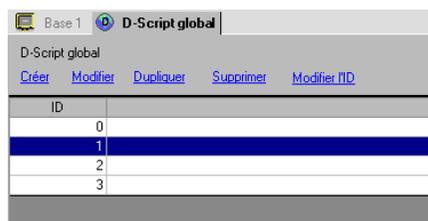
Dans le menu [Objets (P)], sélectionnez [D-Script (R)].

Cliquez sur [Créer]. Pour afficher un script existant, sélectionnez le numéro ID et cliquez sur [Modifier], ou cliquez deux fois sur la ligne du numéro ID.

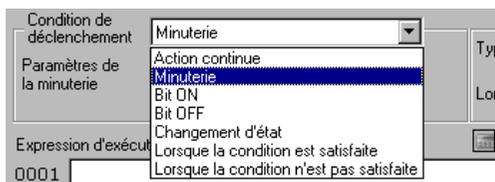


Dans le menu [Outils avancés (R)], sélectionnez [D-Script global (L)].

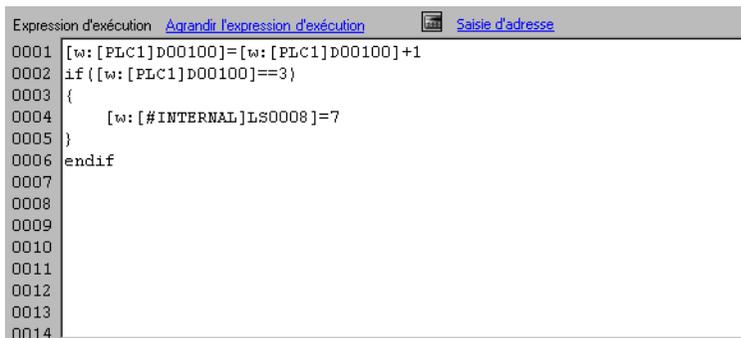
Cliquez sur [Créer]. Pour afficher un script existant, sélectionnez le numéro ID et cliquez sur [Modifier], ou cliquez deux fois sur la ligne du numéro ID.



Configurez la condition de déclenchement qui exécutera le script. Pour plus d'informations sur la présente fonction, reportez-vous à la section «21.8 Configuration de la condition de déclenchement» (page 21-47).



Créez le script (expression d'exécution). Pour plus d'informations sur les commandes et fonctions, reportez-vous à la section «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69).



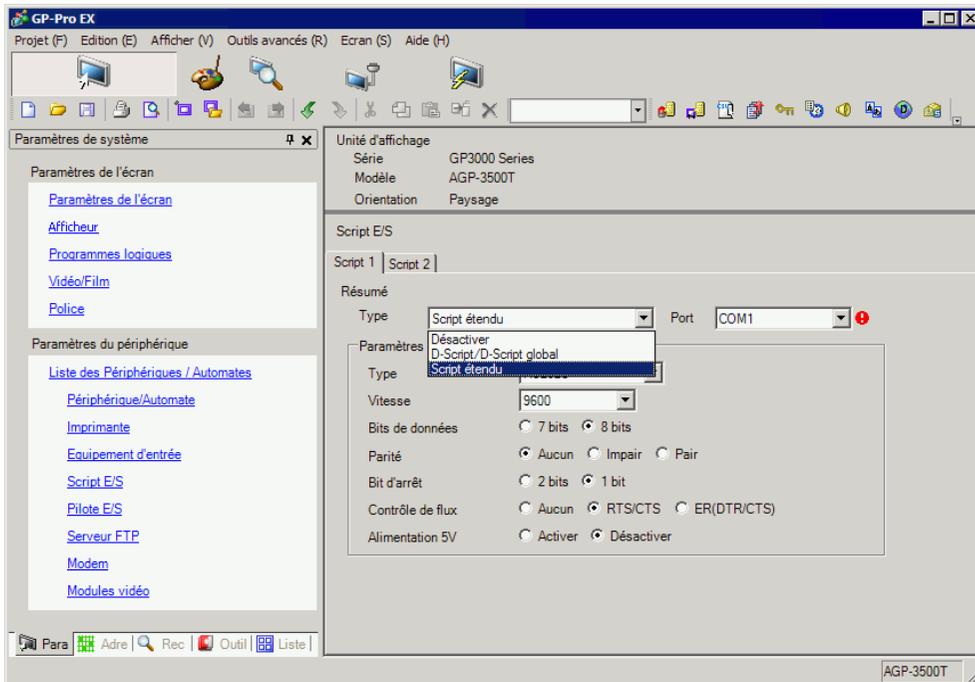
REMARQUE

- Le plateau de composant affiche les D-scripts enregistrés dans l'ordre dans lequel ils ont été créés. Pour modifier l'ordre dans le plateau de composant, modifiez le numéro d'identification des éléments enregistrés, puis dans le menu [Edition], sélectionnez [Aligner les plateaux automatiquement]. Vous pouvez modifier les paramètres d'identification en double-cliquant sur les éléments dans le plateau de composant pour afficher la boîte de dialogue d'édition.
-

21.7.2 Procédure de création des scripts étendus

Dans le menu [Projet (F)], sélectionnez [Paramètres système (C)]. Cliquez sur [Script E/S] pour afficher la boîte de dialogue de configuration suivante.

Si vous utilisez un script étendu, configurez [Script étendu] comme [Type] et sélectionnez le [Port] approprié.



Dans le menu [Outils avancés (R)], sélectionnez [Script étendu (E)].



Créez le script (expression d'exécution). Pour plus d'informations sur les commandes et fonctions, reportez-vous à la section «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69).

```

Expression d'exécution  Agrandir l'expression d'exécution  Saisie d'adresse
0001 [w:[PLC1]D00100]=[w:[PLC1]D00100]+1
0002 if [[w:[PLC1]D00100]=3)
0003 {
0004     [w:[#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
0008
0009
0010
0011
0012
0013
0014
  
```

21.7.3 Configuration des fonctions définies par l'utilisateur

Enregistrez un script existant en tant que fonction définie par l'utilisateur afin de l'utiliser dans d'autres scripts. La fonction enregistrée peut être utilisée par un D-Script, un D-Script global, ou un script étendu.

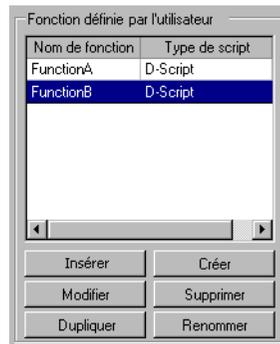
■ Procédure de configuration

Lors de la création d'une nouvelle fonction définie par l'utilisateur

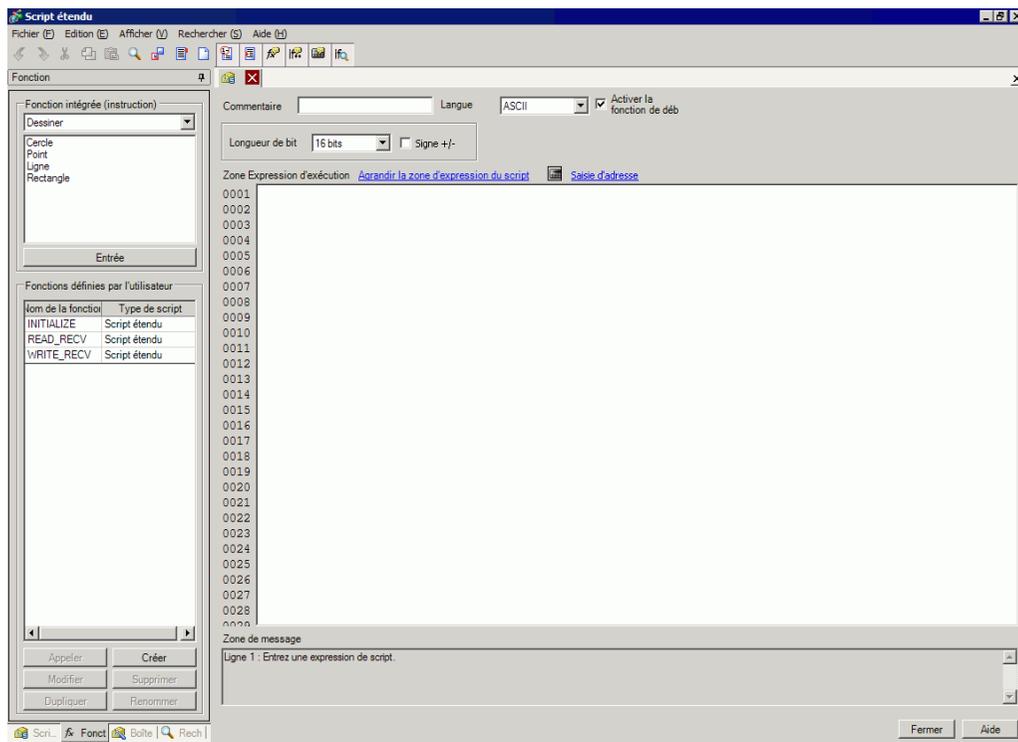
Cliquez sur [Créer]. La boîte de dialogue Fonction définie par l'utilisateur apparaît.

Lors de la modification d'une fonction définie par l'utilisateur enregistrée antérieurement

Sélectionnez la fonction définie par l'utilisateur que vous souhaitez modifier et cliquez sur [Modifier]. La boîte de dialogue Fonction définie par l'utilisateur apparaît.



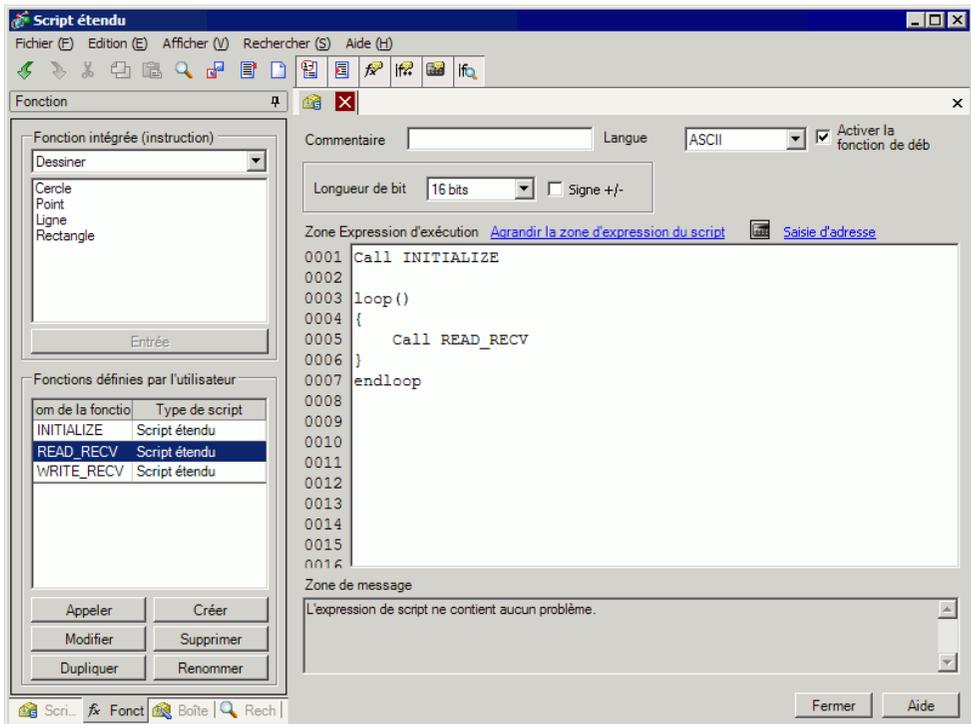
Saisissez le nom de fonction et créez le script dans le champ Exécution. Cliquez sur [OK] pour enregistrer la fonction définie par l'utilisateur.



REMARQUE

- Les restrictions s'appliquent aux noms de fonction. Pour en savoir plus, reportez-vous à la section «21.10.3 Restrictions relatives aux fonctions définies par l'utilisateur» (page 21-66).

Sélectionnez la fonction définie par l'utilisateur à insérer, puis cliquez sur [Insérer] pour placer «Insérer le nom de fonction» dans le champ Exécution.



21.8 Configuration de la condition de déclenchement

Un script créé peut utiliser n'importe lequel des 7 types de conditions de déclenchement suivants :

| Paramètre | | Description |
|-----------------------------|---------------------------------|------------------------------------------------------------------------------------------|
| Action continue | | Le script est déclenché de façon régulière. |
| Minuterie | | Le script est déclenché après qu'une période désignée s'écoule. |
| Bit | Bit ON | Lorsque le GP détecte le bit désigné passe de 0 à 1, le script se déclenche. |
| | Bit OFF | Lorsque le GP détecte le bit défini passant de 1 à 0, le script est déclenché. |
| | Changement de bit | Lorsque le GP détecte le bit désigné passe de 0 à 1 ou de 1 à 0, le script se déclenche. |
| Expressions conditionnelles | Lorsque la condition est vraie | Lorsque le GP détecte qu'une expression désignée est vraie, le script est déclenché. |
| | Lorsque la condition est fausse | Lorsque le GP détecte qu'une expression désignée est fausse, le script est déclenché. |

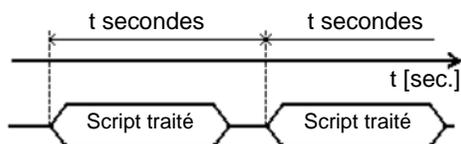
21.8.1 Action continue

S'exécute à chaque temps de cycle.

21.8.2 Minuterie

■ Minuterie

Chaque fois que la durée désignée s'écoule, le script est exécuté une fois. La durée de la minuterie peut être configurée de 1 à 32767 secondes.



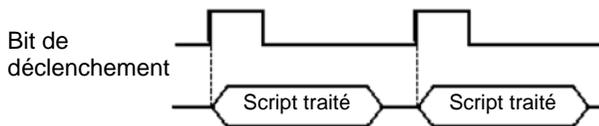
REMARQUE

- Lorsque vous configurez l'heure de la fonction de minuterie, l'heure inclut l'heure configurée + le temps de scrutation d'affichage. De plus, selon le délai pour dessiner un élément d'écran ou pour imprimer des données, il se peut que la fonction de minuterie ralentisse. Pour plus d'informations sur le temps de scrutation d'affichage, reportez-vous à la section « ■ Restrictions relatives au bit déclenché » (page 21-51).
- Lorsque vous utilisez un D-Script, le fait d'alterner l'écran causera la fonction de minuterie de reprendre le compte à partir de 0.

21.8.3 Bit

■ Bit ON

Lorsque le GP détecte l'adresse de bit désignée (bit de déclenchement) passe de 0 à 1, le script se déclenche.

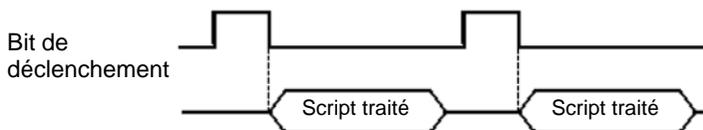


REMARQUE

- Pour l'activation/la désactivation du bit de déclenchement, assurez-vous de laisser passer au moins un intervalle plus long que la période de cycle de communication ou la période de scrutation d'affichage. Pour plus d'informations sur la présente fonction, reportez-vous à la section « ■ Restrictions relatives au bit déclenché » (page 21-51).

■ Bit OFF

Lorsque le GP détecte l'adresse de bit désignée (bit de déclenchement) diminue de 0 à 1, le script est déclenché.

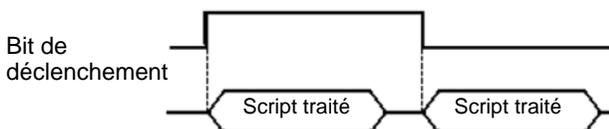


REMARQUE

- Pour l'activation/la désactivation du bit de déclenchement, assurez-vous de laisser passer au moins un intervalle plus long que la période de cycle de communication ou la période de scrutation d'affichage. Pour plus d'informations sur la présente fonction, reportez-vous à la section « ■ Restrictions relatives au bit déclenché » (page 21-51).

■ Changement de bit

Lorsque le GP détecte l'adresse de bit désignée (bit de déclenchement) passe de 0 à 1 ou de 1 à 0, le script est déclenché.



REMARQUE

- Pour l'activation/la désactivation du bit de déclenchement, assurez-vous de laisser passer au moins un intervalle plus long que la période de cycle de communication ou la période de scrutation d'affichage. Pour plus d'informations sur la présente fonction, reportez-vous à la section « ■ Restrictions relatives au bit déclenché » (page 21-51).

21.8.4 Expressions conditionnelles

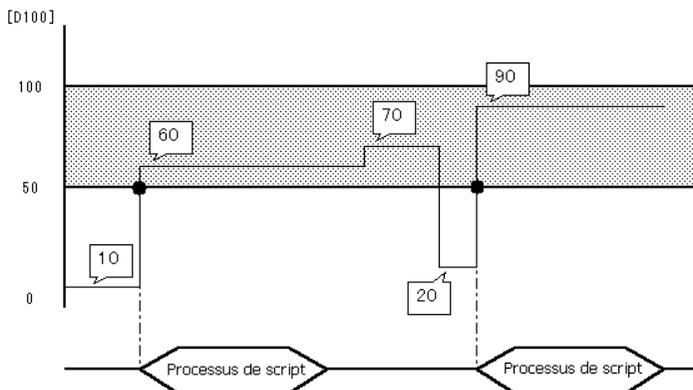
■ Lorsque la condition est vraie

Lorsque le GP détermine que la condition de déclenchement est vraie, le script s'exécute une fois.

Lorsqu'une condition déclenchée est définie sur $100 > [D100] > 50$, le script exécutera la temporisation suivante.

Lorsque [Faux] \rightarrow [Vrai] est détecté, le script s'exécute, et la valeur 70 est assignée à D100.

Le script ne s'exécute pas lorsque [Vrai] \rightarrow [Vrai].



REMARQUE

- Pour la condition déclenchée, laissez passer au moins un intervalle plus long que la période de cycle de communication ou la période de scrutation d'affichage, selon celui de ces délais qui est le plus long. Pour plus d'informations sur la présente fonction, reportez-vous à la section « ■ Restrictions relatives au bit déclenché » (page 21-51).

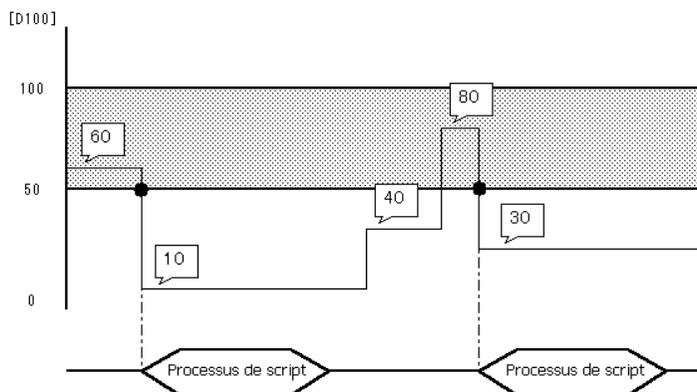
■ Lorsque la condition est fausse

Lorsque le GP détecte qu'une expression désignée dans un programme de déclenchement est fausse, le script est exécuté une fois.

Lorsqu'une condition déclenchée est définie sur $100 > [D100] > 50$, le script exécutera la temporisation suivante.

Lorsque [Faux] \rightarrow [Vrai] est détecté, le script s'exécute, et la valeur 20 est assignée à D100.

Le script ne s'exécute pas lorsque [Vrai] \rightarrow [Vrai].



REMARQUE

- Pour la condition déclenchée, laissez passer au moins un intervalle plus long que la période de cycle de communication ou la période de scrutation d'affichage, selon celui de ces délais qui est le plus long. Pour plus d'informations sur la présente fonction, reportez-vous à la section « ■ Restrictions relatives au bit déclenché » (page 21-51).

■ Restrictions relatives au bit déclenché

- Assurez-vous de laisser passer au moins un intervalle plus long que la période de cycle de communication pour exécuter les opérations d'écriture dans le périphérique connecté. Lorsque des opérations d'écriture sur un périphérique connecté sont exécutées fréquemment à l'aide du compteur de temps de cycle (LS2038), des erreurs de communication ou de système peuvent se produire.
- Si le bit utilisé dans la condition déclenchée du D-Script est défini sur «Bit ON» et que ce bit est désactivé lors du traitement du D-Script, le temps de cycle peut empêcher la détection du front montant. Le déclenchement du D-Script compare la valeur antérieure à la valeur actuelle afin de déterminer si le déclenchement est maintenant «Vrai». Toutefois, lors d'une scrutation simple, la valeur stockée dans l'adresse de bit utilisée lors de l'opération déclenchée demeure la même, même si la valeur est modifiée lors de l'exécution. La nouvelle valeur est lue lors du départ de la prochaine scrutation.

Durée du cycle de communication : Le temps de cycle de communication est la durée pour demander et recevoir des données à partir du périphérique/ automate, jusqu'à ce que l'afficheur reçoive les données. Elle est stockée dans le LS2037 de la variable interne comme données binaires. L'unité est en millisecondes (ms). Il y a une précision de +/-10 ms.

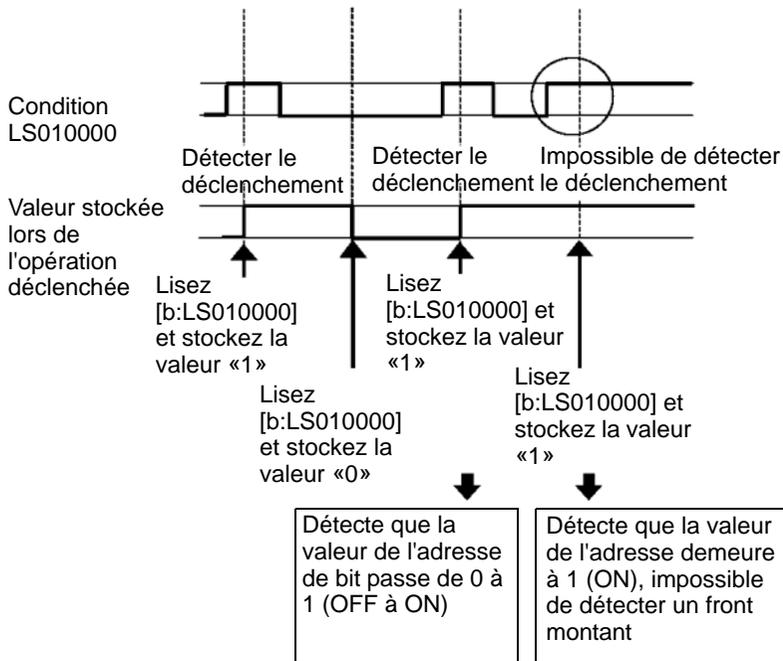
Durée de scrutation de l'affichage : Le temps de scrutation de l'affichage est la durée nécessaire pour traiter un écran. Elle est stockée dans le LS2036 de la variable interne comme données binaires. L'unité est en millisecondes (ms). Il y a une précision de +/-10 ms.

Par exemple, lorsque l'appui est utilisé pour activer le bit de déclenchement (LS010000), et que le D-Script désactive la valeur :

Condition déclenchée : Bit ON [#INTERNAL] LS010000

Expression d'exécution : clear ([b:[#INTERNAL]LS010000])

◆ Chronogramme de traitement du D-Script



Par exemple, si la temporisation d'appui du D-Script n'est pas utilisée, et seule la détection est effectuée, le traitement procédera comme suit :

Utilisation d'une instruction if () pour détecter un déclenchement

Utilisez une instruction «if» pour déterminer si une opération sur appui configure le bit.

Chaque fois que l'instruction if () s'exécute, elle lit la valeur et exécute une vérification de comparaison.

Condition déclenchée : Bit ON ([#INTERNAL]LS203800 *1)

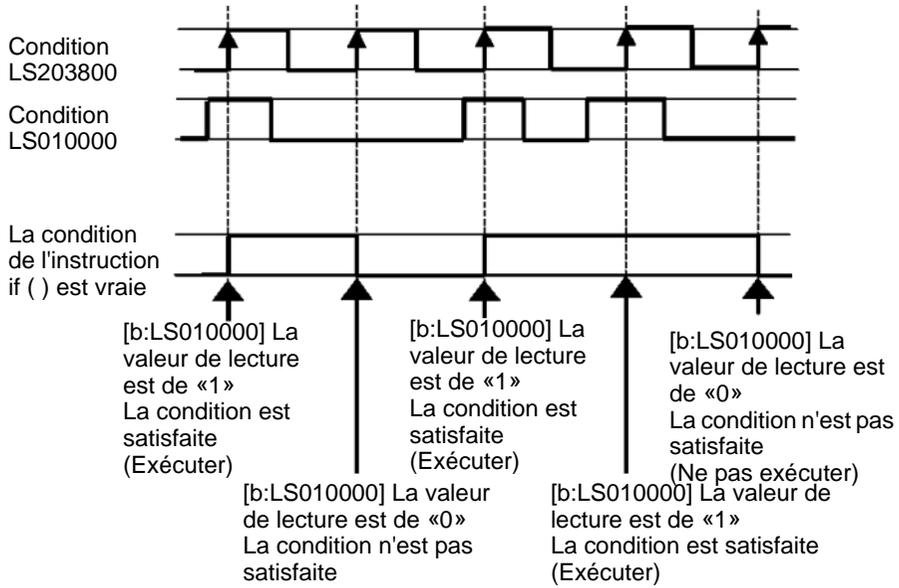
Expression d'exécution : if ([b:[#INTERNAL]LS010000]==1)

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

- *1 Compteur interne du GP Le compteur incrémente chaque fois que l'objet configuré dans l'écran d'affichage est traité.

Lorsque vous utilisez le D-script précédent, même si vous saisissez des appuis consécutifs, le script ne s'exécute que s'il correspond à la condition. Tel qu'indiqué dans le chronogramme suivant, chaque valeur de scrutation d'affichage est lue et vérifiée, et s'il existe une correspondance, peu importe la valeur précédente, le script est exécuté.

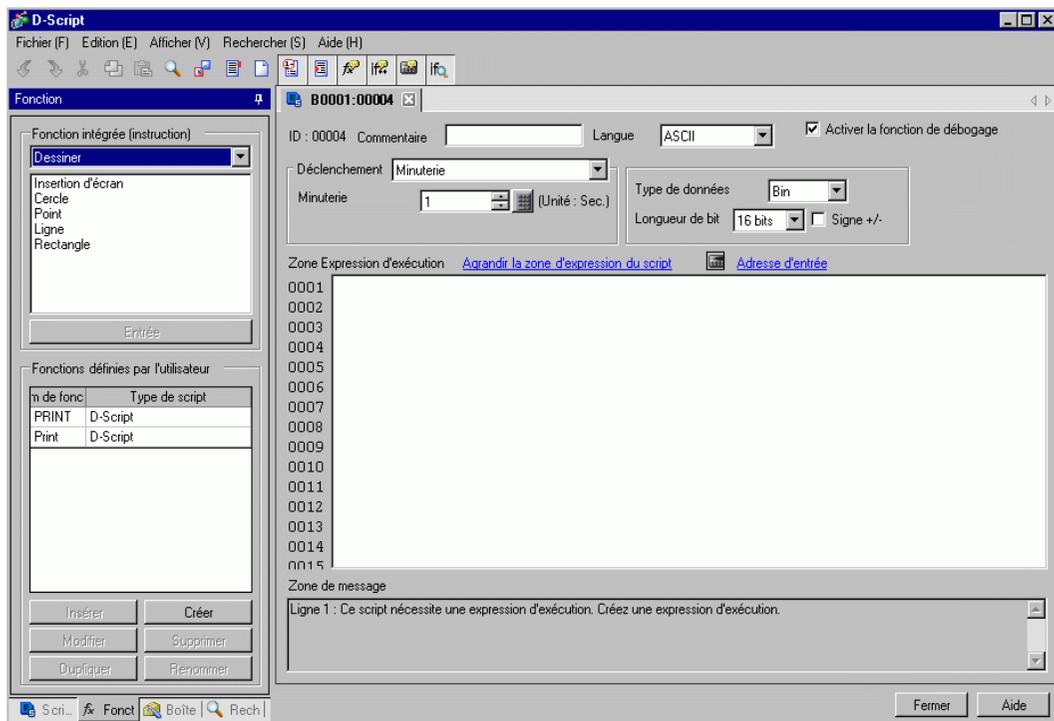
◆ Chronogramme de traitement du D-Script



21.9 Guide de configuration

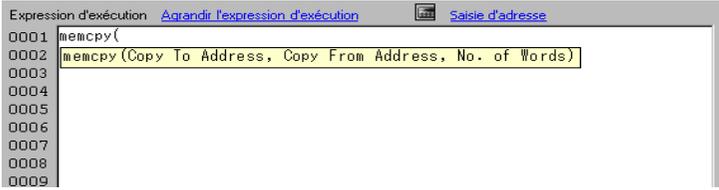
21.9.1 Guide de configuration du D-Script/Outils avancés [D-Script global]

Voici la boîte de dialogue de l'outil avancé [D-Script global]. Les paramètres que vous pouvez préciser pour le D-Script correspondent à ceux dans la boîte de dialogue. Les paramètres d'ID et de déclenchement ne sont pas précisés pour l'outil avancé [Script étendu], toutefois, les autres paramètres sont identiques.

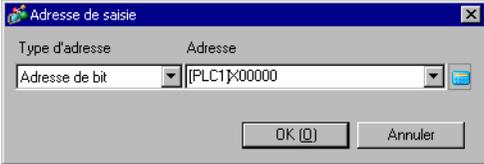


| Paramètre | Description |
|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Exporter | Cela peut être sélectionné à partir du menu Fichier. L'exportation écrit un script créé dans un fichier texte (.txt) qui peut être importé dans d'autres scripts. |
| Importer | Cela peut être sélectionné à partir du menu Fichier. L'importation lit un script exporté (fichier texte). |
| Numéro de ligne  | Affiche le numéro de ligne au côté droit du programme. |
| Contrôle de retrait automatique  | Retrait automatique des instructions, tel que montré ci-dessous. <pre> Expression d'exécution Agrandir l'expression d'exécution Saisie d'adresse 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 0010 0011 0012 </pre> |

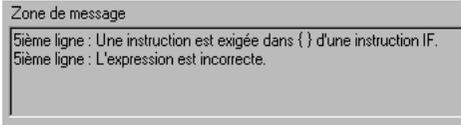
Suite

| Paramètre | Description |
|-----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Aide sur saisie de fonction </p> | <p>Lorsque la fonction et la parenthèse initiale «(» sont saisies telles que montrées ci-dessous, le format de la fonction s'affiche.</p>  |
| <p>Syntaxe automatique terminée </p> | <p>Lorsque «if» ou «loop» est saisi au clavier, la syntaxe restante est automatiquement placée.</p> |

Suite

| Paramètre | Description | | | | | | | | |
|-----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|--|------------------|-------------------|-------------|--|---------------------|-------------------|
| <p>Entrée d'adresse </p> | <p>Lorsque vous créez un script, saisissez un crochet gauche ([) pour afficher la boîte de dialogue [Adresse de saisie].</p>  <p>Sélectionnez le type d'adresse, [Adresse de bit], [Adresse de mot], ou [Adresse temporaire].</p> <ul style="list-style-type: none"> • Adresse de bit Vous pouvez préciser l'adresse de périphérique/automate, Variable interne GP et Variable de bit. • Adresse de mot Vous pouvez préciser l'adresse de périphérique/automate, Variable interne GP et Variable entière. • Adresse temporaire Vous ne pouvez utiliser cette adresse que pour des scripts. <p>Reportez-vous à ce qui suit pour en savoir plus sur les variables internes.</p> <ul style="list-style-type: none"> ☞ «A.1.2 Communication avec un périphérique/automate à l'aide de la méthode d'accès direct» (page A-4) ☞ «A.1.3 Utilisation de la méthode de liaison mémoire avec des périphériques/automates non pris en charge» (page A-7) <p>IMPORTANT</p> <ul style="list-style-type: none"> • Dans les scripts, ne configurez aucun mot de passe qui commence par «0». Toute valeur numérique qui commence avec «0» sera traitée en tant que donnée Oct. (base-8). • Comment décrire des différents formats de saisies de données <p>Par exemple :</p> <ul style="list-style-type: none"> • DEC. (Base-10) : Valeur de départ non nulle Par exemple, 100 • HEX (Base-16) : Valeur commençant par 0x Par exemple, 0x100 • OCT (Base-8) : Valeur commençant par 0x Par exemple, 0100 <ul style="list-style-type: none"> • Exemple d'une opération en formats de données différents à l'aide de l'opérateur AND (Hex et BCD) <table border="0" data-bbox="445 1512 1008 1657"> <tr> <td>Hex. seulement</td> <td></td> </tr> <tr> <td>0x270F et 0xFF00</td> <td>Résultat : 0x2700</td> </tr> <tr> <td>BCD et Hex.</td> <td></td> </tr> <tr> <td>9999 9999 et 0xFF00</td> <td>Résultat : 0x9900</td> </tr> </table> | Hex. seulement | | 0x270F et 0xFF00 | Résultat : 0x2700 | BCD et Hex. | | 9999 9999 et 0xFF00 | Résultat : 0x9900 |
| Hex. seulement | | | | | | | | | |
| 0x270F et 0xFF00 | Résultat : 0x2700 | | | | | | | | |
| BCD et Hex. | | | | | | | | | |
| 9999 9999 et 0xFF00 | Résultat : 0x9900 | | | | | | | | |

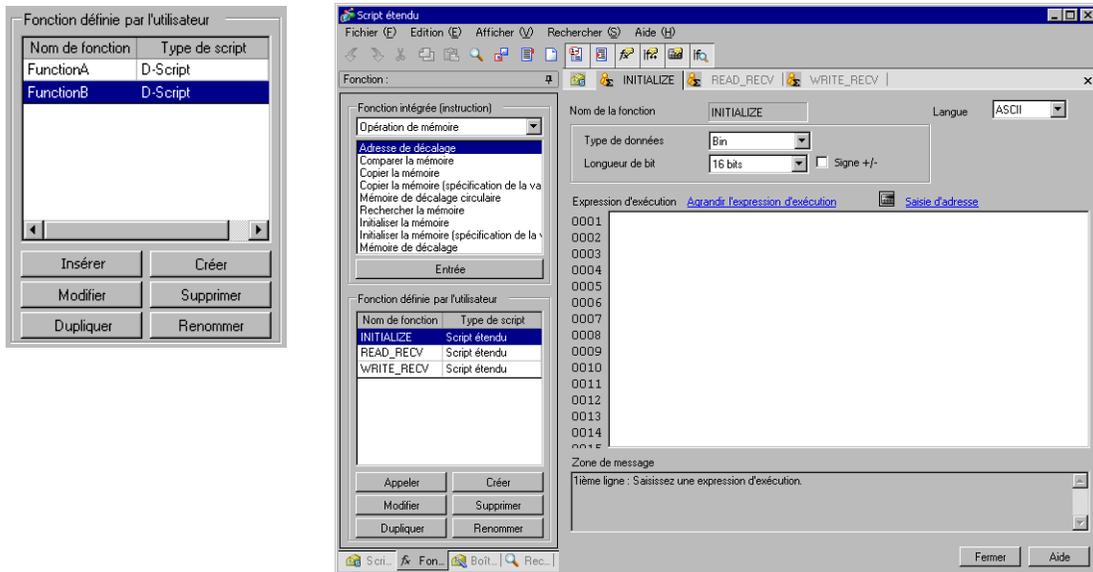
Suite

| Paramètre | Description |
|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Analyse de la syntaxe automatique  | Vérifie la syntaxe lors de la création du script. Les résultats de la vérification seront affichés dans la partie inférieure de la fenêtre.  |
| ID | Les scripts sont gérés par un numéro d'ID. Lorsque vous créez plusieurs scripts ayant des conditions de déclenchement différentes, configurez une valeur comprise entre 0 et 65535. |
| Commentaire | Saisissez un commentaire pour le script. |
| Langue | Choisissez une langue à partir de la liste déroulante : [ASCII], [Japonais], [Chinois (traditionnel)], [Chinois (simplifié)], ou [Coréen]. |
| Activer la fonction de débogage | Configurez s'il faut activer la fonction de débogage. Si la fonction _debug existe dans le script, la fonction _debug s'exécutera. Pour plus d'informations sur la présente fonction, reportez-vous à la section « ■ Fonction de débogage » (page 21-144). |
| Verrouiller le mot de passe | Précisez s'il faut activer la fonction de verrouillage du mot de passe. (1) Cochez la case pour afficher l'écran [Paramètres de niveau d'utilisateur]. (2) Sélectionnez le niveau d'utilisateur parmi [Utilisateur standard], [Grand utilisateur] et [Super utilisateur], puis cliquez sur [OK]. (3) Entrez le [Mot de passe] et cliquez sur [OK]. Lorsque vous cochez la case [Mémoire du mot de passe], le mot de passe est enregistré et la boîte de dialogue Déverrouiller le mot de passe ne s'affiche pas.   |
| Déclenchement | Configurez la condition de déclenchement qui exécutera le script. Pour plus d'informations sur la présente fonction, reportez-vous à la section «21.8 Configuration de la condition de déclenchement» (page 21-47). Les scripts étendus n'ont pas de paramètre de condition de déclenchement. |
| Type de données | Configurez le format des données pour le script à Bin ou BCD. Pour les scripts étendus, le format Bin est fixe. |
| Longueur de bit | Configurez la longueur des données pour le script à 16 bits ou 32 bits. |
| Signe +/- | Sélectionnez cette option lorsque vous souhaitez insérer des numéros négatifs. Vous ne pouvez configurer cette option que si le type de données est Bin. |
| Expression d'exécution | Le contenu du script. |

Suite

| Paramètre | Description |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Fonction intégrée (instruction)</p> | <p>Dans la barre d'outils, sélectionnez des commandes et des fonctions à ajouter facilement au script. Pour plus d'informations sur les commandes disponibles et les fonctions qui peuvent être utilisées, reportez-vous à la section «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69) Fonctions intégrées</p>  <p>Sélectionnez une catégorie à partir de [Fonction intégrée (Instruction)]. Les fonctions reliées apparaissent dans la partie inférieure. Sélectionnez la fonction et cliquez sur [Saisir]. La boîte de dialogue de configuration correspondante apparaît.</p> |
| <p>Fonctions définies par l'utilisateur</p> | <p>Enregistrez un script en tant que fonction définie par l'utilisateur afin de l'utiliser avec d'autres scripts.</p> <p>REMARQUE</p> <ul style="list-style-type: none"> • Pour en savoir plus sur les fonctions définies par l'utilisateur, reportez-vous à la section «21.9.2 Guide de configuration des fonctions définies par l'utilisateur» (page 21-59).  |
| <p>Boîte à outils</p> | <p>Comme raccourci, sélectionnez des commandes à partir de la boîte à outils à utiliser dans le script. De plus, vous pouvez sélectionner des commandes comme un texte de recherche et de position utilisé dans des scripts. Pour plus d'informations sur les commandes disponibles, reportez-vous à la section «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69).</p>  |

21.9.2 Guide de configuration des fonctions définies par l'utilisateur



| Paramètre | Description |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nom de fonction | Affiche les noms des fonctions définies par l'utilisateur. |
| Type de script | Affiche le type de script. Vous pouvez basculer entre [D-Script] et [Script étendu] à l'aide du menu déroulant. |
| Insérer | Insérez une fonction créée. Sélectionnez la fonction à insérer et cliquez sur [Insérer] pour placer la fonction dans le champ Exécution. |
| Créer | Créez une nouvelle fonction. Cliquez sur [Créer]. La boîte de dialogue [Nom de fonction] apparaît. |
| Edition | Modifiez une fonction existante. Sélectionnez la fonction à modifier, puis cliquez sur [Edition]. La boîte de dialogue [Fonction D-script] apparaît. |
| Supprimer | Supprimez une fonction existante. Sélectionnez la fonction à supprimer, puis cliquez sur [Supprimer]. |
| Dupliquer | Copiez une fonction existante. Sélectionnez la fonction à copier et cliquez sur [Copier] pour afficher la boîte de dialogue et créer le nom de la copie. |
| Renommer | Modifiez le nom d'une fonction existante. Cliquez sur [Renommer]. La boîte de dialogue Renommer la fonction apparaît. |

21.10 Restrictions

21.10.1 Restrictions relatives au D-Script/D-Script global

- Dans la programmation de D-Scripts, trois adresses utilisent autant de mémoire qu'un objet. Le nombre maximum d'adresses disponibles pour un D-script est de 255^{*1} . Utilisez le moins d'adresses possibles, car plus il y a de périphériques, plus le temps de réponse est long.
- Le D-Script ne peut pas exécuter des calculs sur les valeurs flottantes (variables flottantes ou variables réelles). Vous ne pouvez pas non plus exécuter des calculs sur les variables structurées. Toutefois, vous pouvez exécuter des calculs sur les éléments individuels à partir des variables de structure.
- La taille d'un D-Script a un effet sur la durée de scrutation de l'affichage. Remarquez que si vous utilisez un grand nombre d'adresses, il se peut que la performance du programme soit diminuée de manière significative.
- Ne précisez pas l'[Action continue] dans les conditions de déclenchement pour le script à écrire dans les adresses de périphérique/automate. Une erreur s'affichera car le traitement de communication ne peut pas conserver toutes les instructions d'écriture. Pour activer l'[Action continue], utilisez la variable interne du GP ou une adresse temporaire.
- Lorsque vous insérez une sous-fonction, le nombre maximum de niveaux imbriqués est de 9. Assurez-vous de ne pas en créer plus de 9.
- Vous pouvez créer jusqu'à 9 niveaux d'appels imbriqués.
- Vous pouvez créer jusqu'à 254 fonctions.

*1 Le nombre total de périphériques utilisés dans les expressions de déclenchement et les programmes de script.

◆ Selon les périphériques spécifiés pour les conditions de déclenchement, les opérations de D-script activées par un déclenchement après que l'écran est modifié sont les suivantes :

| Condition de déclenchement | Tout périphérique connecté autre que [#MEMLINK] | | | | [#MEMLINK] | | | |
|----------------------------|-------------------------------------------------|---------|---------|--------------------------|----------------------|---------|---------|--------------------------|
| | Valeur ou condition actuelle | Bit «0» | Bit «1» | Condition pas satisfaite | Condition satisfaite | Bit «0» | Bit «1» | Condition pas satisfaite |
| Bord montant | X | O | – | – | X | X | – | – |
| Bord descendant | O | X | – | – | X | X | – | – |
| Changement de bit | O | O | – | – | X | X | – | – |
| Paramètres de la minuterie | X | X | X | X | X | X | X | X |
| Détection de «Vrai» | – | – | X | O | – | – | X | O |
| Détection de «Faux» | – | – | O | X | – | – | O | X |

O : L'opération est effectuée immédiatement après que l'écran est modifié, ou après que la machine est mise sous tension.

X : L'opération n'est pas effectuée immédiatement après que l'écran est modifié, ou après que la machine est mise sous tension.

- Si la minuterie est en cours d'exploitation, elle lance le compte immédiatement après que l'écran est modifié.
- Lorsque vous utilisez un D-Script global, les opérations mentionnées ci-dessus ne sont effectuées que si le GP est mis sous tension. Toutefois, lorsque l'écran GP est modifié, l'opération précédente n'est pas effectuée et les conditions de déclenchement sont surveillées de façon répétitive.
- Lorsqu'un D-Script global comprend une minuterie, la minuterie lance le compte immédiatement après que le GP est mis sous tension.

REMARQUE

- N'utilisez pas une touche pour configurer le bit de déclenchement ou pour activer le bit de départ dans un programme. Il se peut que la temporisation de la saisie sur appui ne soit pas correcte, ce qui engendre la saisie incorrecte du bit.

- ◆ **Lorsqu'une valeur est affectée à une adresse pour basculer l'écran lorsqu'une commande D-Script est en cours d'exécution, l'opération de basculement de l'écran est traitée après que tous les D-Scripts sont traités.**

Par exemple :

```

ID          00000
Type de données Bin          Longueur des données 16 bits Signe +/- Aucun
Déclenchement Bit de poids fort ([b:M0000])
[w:[PLC1]D0100]=0           // (1)
[w:[#INTERNAL]LS0008]=30    // (2) Bascule au n° d'écran de base 30
[w:[PLC1]D0101]=1          // (3)
[w:[PLC1]D0102]=2          // (4)

```

Lorsque le D-Script précédent est exécuté, le traitement du basculement d'écran est effectué après que les étapes (3) et (4) ont été traitées.

- ◆ **Lorsqu'une opération sur appui GP est configurée sur des données utilisées dans un D-script, assurez-vous que l'opération d'écriture de données se termine avant d'exécuter le D-script.**

◆ Restrictions spécifiques au D-Script global

- Lorsque le GP est mis sous tension, les actions indiquées dans le tableau de la page précédente sont effectuées. Lors du changement de l'écran, le tableau ci-dessus n'est pas appliqué et les conditions de déclenchement sont contrôlées de façon continue.
- L'opération du D-Script global est suspendue lors du changement de l'écran ou d'autres opérations GP.
- Après que le GP est mis sous tension, les actions du D-Script global ne sont pas effectuées avant que la lecture de toutes les données soit terminée pour l'écran initial. Après que l'écran initial est modifié, les actions du D-Script global peuvent être effectuées avant que la lecture des données soit terminée.
- Le nombre maximum de périphériques dans les D-scripts globaux est de 255^{*1}. Si le nombre maximum est atteint, le D-script ne fonctionne pas. Puisque ces périphériques lisent toujours les données peu importe l'écran, assurez-vous de n'utiliser que le nombre minimum de paramètres de périphérique dans votre D-Script. Sinon, il se peut que la performance de l'opération soit diminuée.
- Le nombre maximum de D-Scripts globaux disponibles est de 32. La fonction en cours d'utilisation compte également comme un D-Script global. Lorsque le nombre de D-Scripts globaux atteint 32, tout autre D-Script global est ignoré.

*1 Le nombre total de périphériques utilisés dans les expressions de déclenchement et les programmes de script.

◆ Restrictions relatives aux opérations du port SIO

- Les adresses désignées dans les fonctions d'envoi ou de réception ne sont pas ajoutées aux adresses D-Script.
- Le contrôle est une variable en écriture seule, tandis que les données de statut et de réception sont des variables en lecture seule. Ne tentez pas de lire la variable de contrôle ou d'écrire des données dans la variable de statut, ou l'opération échouera.
- Créez des D-Scripts (ou des fonctions) pour les opérations d'envoi et de réception. Pour plus d'informations sur les organigrammes des transferts de données, reportez-vous à la section
 « ■ Organigramme » (page 21-24)
- La zone utilisateur dans le périphérique LS (de LS20 à LS2031 et de LS2096 à LS8191) peut stocker des données pour les fonctions d'envoi ou de réception.
- Dans l'espace de travail [Paramètres système], page [Script E/S], si [D-Script/D-Script global] n'est pas configuré comme [Type], le 13e bit dans l'adresse LS2032 s'active lorsque le [D-Script/D-Script global] exécute les fonctions d'étiquette de l'[Opération de port SIO] (Envoyer, Recevoir, Contrôler, Statut de lecture et Taille des données de réception). Pour plus d'informations sur les relais spéciaux :
 « A.1.4.3 Relais spécial » (page A-25)
- Lorsque vous utilisez des fonctions d'envoi ou de réception, configurez la longueur de bit du D-Script à 16 bits. Remarquez que l'opération échoue si la longueur du bit est configurée à 32 bits.
- La taille du tampon d'envoi est de 2048 octets, tandis que le tampon de réception est de 8192 octets. Les signaux ER (sortie) et RS (sortie) sont désactivés après qu'au moins 80 % du tampon de réception est rempli.

◆ Restrictions relatives aux opérations de format BCD

Si une valeur qui ne peut pas être convertie en format BCD est trouvée lors de l'opération, le programme arrête l'exécution.

Ces valeurs incluent de A à F en format hexadécimal.

N'utilisez pas de telles valeurs. Si le programme s'arrête en raison des valeurs non BCD, le bit 7 dans les informations de relais communes (LS2032) sur le GP est activé. Ce bit ne se désactive pas avant que le GP soit mis hors tension ou hors ligne.

Par exemple :

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

Si D300 est 3, le décalage de deux bits vers la gauche produit 0x000C, ce qui ne peut pas être converti en format BCD et l'exécution du programme est interrompue.

$[w:[PLC1]D0200]=[w:[PLC1]D0300]<<2$ Si D300 est 3, le décalage de deux bits vers la gauche produit 0x000C.

A la différence de l'exemple précédent, 0x000C est le résultat de l'opération à stocker dans la mémoire et n'arrête pas le programme.

◆ Restrictions relatives aux opérations nulles

Si vous divisez par zéro dans des opérations de division (/) et de module (%), l'exécution s'arrêtera. Ne divisez pas par zéro.

Si le programme s'arrête suite à l'erreur ci-dessus, le bit 8 dans les informations de relais communes (LS2032) sur le GP s'active. Ce bit ne se désactive pas avant que le GP soit mis hors tension ou hors ligne.

◆ Remarques concernant le délai pendant l'opération d'affectation

L'utilisation d'une adresse de périphérique lors d'une opération d'affectation peut produire un délai d'écriture puisque le GP doit lire les données d'adresse à partir du périphérique connecté. Considérez ce qui suit :

Par exemple :

```
[w:[PLC1]D0200]=[w:[PLC1]D0300]+1 ...
```

```
[w:[PLC1]D0201]=[w:[PLC1]D0200]+1 ...
```

L'instruction (1) affecte (D0300+1) dans D0200. Toutefois, dans l'instruction (2), le résultat de l'instruction (1) n'a pas été affecté dans D0200 en raison du temps de cycle de l'automate. Dans un tel cas, programmez le script pour que le résultat de l'instruction (1) soit stocké dans la zone LS avant qu'il soit exécuté, tel que montré ci-dessous.

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

```
[w:[PLC1]D0200] = [w:[#INTERNAL]LS0100]
```

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

◆ Remarques sur comment traiter les nombres négatifs

Pour les fonctions pour lesquelles un nombre négatif est entré pour un argument qui n'accepte pas les nombres négatifs*1, le nombre entré est non signé*2.

*1 Par exemple, «le nombre d'octets» de l'argument `_CF_read ()` n'accepte pas les nombres négatifs car il s'agit de la taille des données à lire.

*2 Par exemple, -1 est traité comme 65535 pour 16 bits, et 4294967295 pour 32 bits.

21.10.2 Restrictions relatives aux scripts étendus

- Pour les adresses de périphérique, seules les zones LS et USR (zone utilisateur étendue) peuvent être utilisées.
- Les adresses temporaires des D-Scripts et des D-Scripts globaux sont gérées de façon indépendante à partir de l'adresse temporaire des scripts étendus. Donc, les changements effectués aux adresses temporaires des D-Scripts et des D-Scripts globaux ne sont pas reflétés dans l'adresse temporaire des scripts étendus.
- Vous pouvez insérer des fonctions définies par l'utilisateur créées à l'aide d'un D-Script/D-Script global, mais si vous accédez à une adresse de périphérique à l'extérieur de la plage des variables internes de la fonction, il se peut qu'elles ne fonctionnent pas correctement. De plus, lors du transfert (lors de la création des données pour le GP), les fonctions définies par l'utilisateur sont créées de façon indépendante pour les D-Scripts, les D-Scripts globaux, et les scripts étendus.
- Lorsque vous insérez une fonction à partir d'une fonction, le nombre maximum de niveaux imbriqués est de 9.
- Vous pouvez insérer jusqu'à 254 fonctions. (Le nombre de fonctions disponibles avec l'option «Insérer» est de 254.)
- Le script étendu n'a aucun effet sur le nombre de balises.
- Les fonctions qui ne sont prises en charge que par des scripts étendus, par exemple les opérations de chaîne, ne fonctionnent pas si elles sont insérées à l'aide d'un D-Script ou d'un D-Script global.
- Le format de données disponible est Bin. Le format de données BCD est désactivé.
- La taille du tampon d'envoi est de 2048 octets, tandis que le tampon de réception est de 8192 octets. La ligne CTS est désactivée après qu'au moins 80 % du tampon de réception est rempli.
- Vous ne pouvez pas sélectionner un D-script/D-script global et un script étendu simultanément. Remarquez les combinaisons énumérées dans le tableau ci-dessous.

| Paramètre SIO étendu | D-Script/D-Script global | Fonction SIO étendu pour un script étendu |
|--------------------------|-------------------------------|-------------------------------------------|
| D-Script/D-Script global | O : Fonctionnement possible | X : Fonctionnement impossible |
| Script étendu | X : Fonctionnement impossible | O : Fonctionnement possible |

- Conventions notationnelles pour le paramètre de chaîne de caractères

Lorsque vous utilisez des chaînes de caractères avec «_strset ()» et d'autres fonctions, encadrez la chaîne de caractères entre guillemets ("). Pour afficher les guillemets dans les chaînes de caractères, ajoutez le signe «\» et exprimez-le comme [\"]. Il n'existe aucune façon pour représenter un signe «\» unique. Au besoin, utilisez le paramètre de format du code de caractères (_strset (databuf0, 92)).

Par exemple :

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

- Pour les fonctions pour lesquelles un nombre négatif est entré pour un argument qui n'accepte pas les nombres négatifs^{*1}, le nombre entré est non signé^{*2}.

*1 Par exemple, «le nombre d'octets» de l'argument _CF_read () n'accepte pas les nombres négatifs car il s'agit de la taille des données à lire.

*2 Par exemple, -1 est traité comme 65535 pour 16 bits, et 4294967295 pour 32 bits.

◆ Tailles des tampons SIO détaillé spécialisés (databuf0, databuf1, databuf2 et databuf3)

| Tampon | Nom de tampon | Taille |
|---------------------|---------------|--------|
| Tampon de données 0 | databuf0 | 1 Ko |
| Tampon de données 1 | databuf1 | 1 Ko |
| Tampon de données 2 | databuf2 | 1 Ko |
| Tampon de données 3 | databuf3 | 1 Ko |

21.10.3 Restrictions relatives aux fonctions définies par l'utilisateur

- Les parties des commandes que l'on peut utiliser diffèrent selon chaque script. Lorsque vous utilisez les commandes, reportez-vous à la section «21.11 Commandes de programme/Expressions conditionnelles» (page 21-69).
- Pour le nom de fonction, vous pouvez utiliser toutes les lettres françaises ou le caractère de soulignement «_». (Cependant, le nom de fonction doit commencer par un caractère alphanumérique.)
- N'utilisez pas ce qui suit comme noms de fonction :

| | | | | | |
|-------------|--------------|------------|---------------|-------------|---------------|
| and | b_call | Bcall | _bin2hexasc | break | Insérer |
| _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 Remarques sur les résultats d'opération

■ Chiffres superflus

Tout chiffre superflu résultant des opérations est tronqué.

Lorsque vous effectuez une opération sur des données 16 bits non signées :

- $65535 + 1 = 0$ (Produit des chiffres superflus)
- $(65534 * 2) / 2 = 32766$ (produit des chiffres superflus)
- $(65534 / 2) * 2 = 65534$ (Ne produit aucun chiffre superflu)

■ Différence de traitement résiduel

Le résultat d'un traitement résiduel est déterminé si les côtés gauche et droit sont signés ou non signés.

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

■ Décimales tronquées

Les valeurs fractionnaires résultant de l'opération sont tronquées.

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

■ Remarques concernant l'exploitation des données BCD

Une opération de données BCD qui produit des chiffres superflus ne donne pas le bon résultat.

21.10.5 Erreurs

Le message d'erreur suivant s'affiche lorsque vous ne configurez pas un script correctement. L'erreur s'affichera dans la partie inférieure de l'écran GP.

Les codes d'erreur sont écrits dans les adresses LS91XX. Le numéro écrit dans la zone de code d'erreur sera la partie du numéro qui suit RAAA dans le tableau suivant. (Par exemple, lorsque l'erreur RAAA130 se produit, «130» sera écrit.)

Liste de codes d'erreur de script

| D-Script (Adresse d'erreur=LS9120) | D-Script global (Adresse d'erreur=LS9110) | Script étendu (Adresse d'erreur=LS9100) |
|---------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| – | RAAA130 | RAAA140 |
| Inutilisé | Erreur de D-Script global. (Le nombre total de D-Scripts globaux dépasse la limite de 32.) | Erreur de D-Script global (le nombre total de fonctions dépasse la limite de 255.) |
| – | RAAA131 | – |
| Inutilisé | Erreur de D-Script global. (Le nombre total de périphériques dépasse la limite de 255 ^{*1} .) | Inutilisé |
| RAAA120 | RAAA132 | RAAA141 |
| Erreur D-Script (la fonction précisée n'existe pas ou la fonction contient une erreur.) | Erreur D-Script global (la fonction précisée n'existe pas ou la fonction contient une erreur.) | Erreur D-Script étendu (la fonction précisée n'existe pas ou la fonction contient une erreur.) |
| RAAA121 | RAAA133 | RAAA142 |
| Erreur D-Script (ces fonctions sont imbriquées dans 10 niveaux ou plus.) | Erreur D-Script global (ces fonctions sont imbriquées dans 10 niveaux ou plus.) | Erreur D-Script étendu (ces fonctions sont imbriquées dans 10 niveaux ou plus.) |
| RAAA122 | RAAA134 | RAAA143 |
| Erreur D-Script (une expression existe, qui n'est pas prise en charge par cette version.) | Erreur D-Script global (une expression existe, qui n'est pas prise en charge par cette version.) | Erreur D-Script étendu (une expression existe, qui n'est pas prise en charge par cette version.) |
| RAAA123 | RAAA135 | RAAA144 |
| Erreur D-Script (la fonction d'opération SIO est utilisée dans une condition pour laquelle aucun périphérique/ automate n'a été configuré.) | Erreur D-Script global (la fonction d'opération SIO est utilisée dans une condition pour laquelle aucun périphérique/ automate n'a été configuré.) | Erreur D-Script étendu (la fonction d'opération SIO est utilisée dans une condition pour laquelle aucun périphérique/ automate n'a été configuré.) |
| RAAA124 | RAAA136 | RAAA145 |
| Le D-Script contient une erreur. | Le D-Script global contient une erreur. | Le D-Script étendu contient une erreur. |

*1 Le nombre total de périphériques utilisés dans les expressions de déclenchement et les programmes de script.

21.11 Commandes de programme/Expressions conditionnelles

■ trigonométrie

| Élément | Commande/Fonction | D-Script/D-Script global | Script étendu |
|-----------------|----------------------------|--------------------------|---------------|
| Type de données | Bin, BCD | O | Bin seulement |
| Longueur de bit | 16 bits, 32 bits | O | O |
| Signé +/- | Activé/Désactivé | O | O |
| Déclenchement | Paramètres de la minuterie | O | X |
| | Bord montant du bit | O | X |
| | Bord descendant du bit | O | X |
| | Bit activé/désactivé | O | X |
| | L'expression est vraie | O | X |
| | L'expression est fausse | O | X |
| Dessiner | Ecran de chargement | O | X |
| | Point | O | O |
| | Ligne | O | O |
| | Cercle | O | O |
| | Rectangle | O | O |
| Opérateur | Addition (+) | O | O |
| | Soustraction (-) | O | O |
| | Module (%) | O | O |
| | Multiplication (*) | O | O |
| | Division (/) | O | O |
| | Affectation (=) | O | O |
| Comparaison | ET logique | O | O |
| | OU logique | O | O |
| | Négation (non) | O | O |
| | Inférieur à (<) | O | O |
| | Inférieur ou égal à (<=) | O | O |
| | Pas égal à (<>) | O | O |
| | Supérieur à (>) | O | O |
| | Supérieur ou égal à (>=) | O | O |
| | Egal à (==) | O | O |

Suite

| Élément | Commande/Fonction | D-Script/D-Script global | Script étendu |
|-----------------------------|-------------------------------------------------------------------|--------------------------|------------------|
| Opération de mémoire | Copie de mémoire : memcpy () | O | O |
| | Initialiser la mémoire : memset () | O | O |
| | Copie de mémoire (spécification de variable) _memcpy_EX () | O | O |
| | Initialiser la mémoire (spécification de variable) _memset_EX () | O | O |
| | Adresse de décalage | O | O |
| | Décaler la mémoire | O | O |
| | Memory Ring | O | O |
| | Rechercher la mémoire | O | O |
| | Comparer la mémoire | O | O |
| Opération sur bit | Décalage gauche (<<) | O | O |
| | Décalage droit (>>) | O | O |
| | Bitwise AND (&) | O | O |
| | Bitwise OR () | O | O |
| | Bitwise XOR (^) | O | O |
| | Complément à 1 | O | O |
| | Configurer le bit : set () | O | O |
| | Effacer le bit : clear () | O | O |
| | Bit activé/désactivé : toggle () | O | O |
| Expressions conditionnelles | if () | O | O |
| | if () else | O | O |
| | loop (), break | O | O |
| | loop () boucle infinie | X | O |
| Adresse | Adresse de bit | O | Variable interne |
| | Adresse de mot | O | Variable interne |
| | Adresse de fonctionnement temporaire | O | O*1 |
| Constante | Déc., Hex, Oct. | O | O |

Suite

| Élément | Commande/Fonction | D-Script/D-Script global | Script étendu |
|--------------|------------------------------------------------------|--------------------------|---------------|
| Fonction SIO | Recevoir : IO_READ ([p:SIO]) | O | O |
| | Envoyer : IO_WRITE ([p:SIO]) | O | O |
| | Réception étendue : _IO_READ_EX () | X | O |
| | Envoi étendu _IO_WRITE_EX () | X | O |
| | Fonction de réception en veille _IO_READ_WAIT () | X | O |
| | Contrôle [c:EXT_SIO_CTRL] | O | O |
| | Statut [s:EXT_SIO_STAT] | O | O |
| | Taille des données reçues [r:EXT_SIO_RCV] | O | O |
| | Pause : _wait () | X | O |

| Élément | Commande/Fonction | D-Script/D-Script global | Script étendu |
|--------------------|--------------------------------------------------------------------------------|--------------------------|---------------|
| Opération de texte | Texte | X | O |
| | Tampon de données databuf0, databuf1, databuf2, databuf3 | X | O |
| | Chaîne d'écriture _strset () | X | O |
| | Depuis le tampon de données vers la variable interne _dlcopy () | X | O |
| | Copier à partir de la variable interne vers le tampon de données : _ldcopy () | X | O |
| | Conversion Text hexadécimal -> Entier _hexasc2bin () | X | O |
| | Conversion Text décimal -> Entier _decasc2bin () | X | O |
| | Conversion Nombre hexadécimal -> Chaîne _bin2hexasc () | X | O |
| | Conversion Nombre décimal -> Chaîne _bin2decasc () | X | O |
| | Longueur de chaîne _strlen () | X | O |
| | Concaténation de chaîne _strcat () | X | O |
| | Copier les chaînes partielles _strmid () | X | O |
| | Statut [e:STR_ERR_STAT] | X | O |

Suite

| Élément | Commande/Fonction | D-Script/D-Script global | Script étendu |
|--------------------------|-----------------------------------------------|--------------------------|---------------|
| Fonction | Insérer | O | O |
| | return | X | O |
| Opération de fichier CF | Lire le fichier CSV | O | O |
| | Produire la liste de fichiers _CF_dir () | O | O |
| | Lire le fichier _CF_read () | O | O |
| | Lire le fichier CSV CF_read_csv () | O | O |
| | Ecrire le fichier _CF_write () | O | O |
| | Supprimer le fichier _CF_delete () | O | O |
| | Modifier le nom de fichier _CF_rename () | O | O |
| Opération de fichier USB | Fichier de lecture USB | O | O |
| | Produire la liste de fichiers _USB_dir () | O | O |
| | Lire le fichier _USB_read () | O | O |
| | Lire le fichier CSV _USB_read_csv () | O | O |
| | Ecrire le fichier _USB_write () | O | O |
| | Supprimer le fichier _USB_delete () | O | O |
| | Modifier le nom de fichier _USB_rename () | O | O |
| Opération d'imprimante | Port COM de sortie IO_WRITE ([p:PRN]) | O | O |
| Déboguer | _debug () | O | O |

*1 L'adresse temporaire existe de façon indépendante du D-Script et du D-Scripts global.

21.11.1 Opération sur bit

| Opération sur bit | Résumé de la fonction |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Paramètres de bit  « ■ Paramètres de bit » (page 21-74) Modifie l'adresse de bit précisée de 0 à 1. |
| | Effacer le bit  « ■ Effacer le bit » (page 21-74) Modifie l'adresse de bit précisée de 1 à 0. |
| | Bit activé/désactivé  « ■ Bit activé/désactivé » (page 21-74) Modifie l'adresse de bit spécifiée de 1 à 0 ou de 0 à 1. |

■ Paramètres de bit

| Élément | Description |
|---------|---------------------------------------------|
| Résumé | Modifie l'adresse de bit précisée de 0 à 1. |
| Format | set () |

Exemple d'expression :

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

Dans l'exemple ci-dessus, le le bit 00 du LS0100 passe de 0 à 1.

■ Effacer le bit

| Élément | Description |
|---------|---------------------------------------------|
| Résumé | Modifie l'adresse de bit précisée de 1 à 0. |
| Format | clear () |

Exemple d'expression :

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

Dans l'exemple ci-dessus, le le bit 00 du LS0100 passe de 1 à 0.

■ Bit activé/désactivé

| Élément | Description |
|---------|----------------------------------------------------------|
| Résumé | Modifie l'adresse de bit spécifiée de 1 à 0 ou de 0 à 1. |
| Format | toggle () |

Exemple d'expression :

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

Dans l'exemple ci-dessus, le bit 00 du LS0100 passe de 1 à 0 ou de 0 à 1.

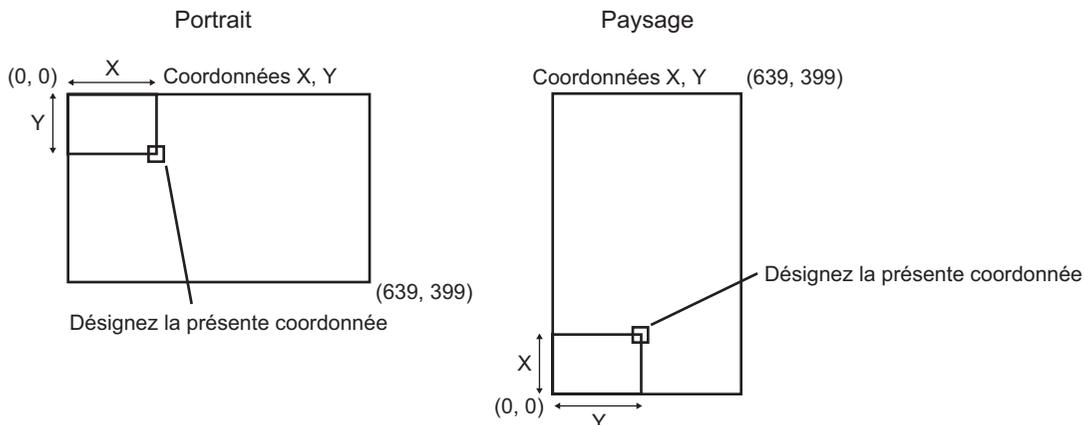
21.11.2 Dessiner

| Dessiner | Résumé de la fonction |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Ecran d'insertion « ■ Ecran d'insertion » (page 21-75) Insère l'écran (écran de base) avec le numéro d'écran désigné. Il est impossible de l'utiliser dans un script étendu.</p> |
| | <p>Cercle « ■ Ligne » (page 21-77) Dessine le cercle désigné.</p> |
| | <p>Point « ■ Point » (page 21-77) Dessine le point désigné.</p> |
| | <p>Ligne « ■ Ligne » (page 21-77) Dessine la ligne désignée.</p> |
| | <p>Rectangle « ■ Rectangle » (page 21-78) Dessine le rectangle désigné.</p> |

■ Ecran d'insertion

| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Cette fonction insère un élément de bibliothèque. L'écran désigné (écran de base) est inséré aux coordonnées X et Y désignées. Il est impossible de l'utiliser dans un script étendu.</p> |
| Format | <p>b_call (Numéro d'écran, Coordonnée en X, Coordonnée en Y)</p> <div style="text-align: center;">  </div> <p>REMARQUE</p> <ul style="list-style-type: none"> • Configurez la coordonnée du centre de l'écran avec la coordonnée en X et le coordonnée en Y. |

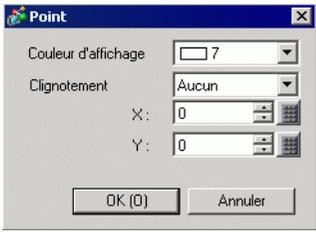
Position de la coordonnée



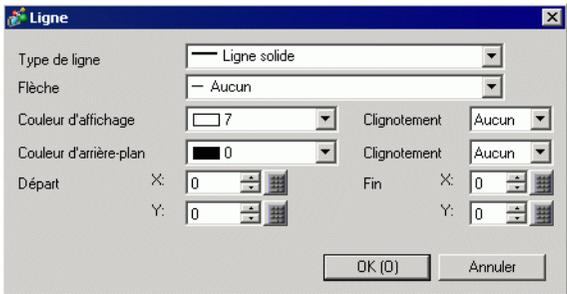
■ Cercle

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Dessine un cercle au point désigné. Lorsque vous cochez la case [Motif], un cercle rempli est dessiné. Sélectionnez et saisissez le type de ligne (ou le motif du remplissage lorsque vous sélectionnez un motif), les attributs de la couleur, les coordonnées du centre et la valeur du rayon. Les coordonnées du centre et le rayon peuvent être configurés de façon indirecte. |
| Format | <p>dsp_cercle (Coordonnée en X, Coordonnée en Y, Rayon, Clignotement de la couleur d'affichage + Couleur d'affichage, Clignotement de la couleur de l'arrière-plan + Couleur de l'arrière-plan, Type de ligne)</p> <div data-bbox="548 1136 1090 1425" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>REMARQUE</p> <ul style="list-style-type: none"> • Lorsque la couleur noire et le clignotement sont configurés, la couleur de l'arrière-plan devient transparente. |

■ Point

| Elément | Description |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Dessine un point au point désigné. Configurez les coordonnées en X et Y, et la couleur d'affichage. |
| Format | <p>dsp_dot (Coordonnée en X, Coordonnée en Y, Clignotement + Couleur d'affichage)</p>  <p>REMARQUE</p> <ul style="list-style-type: none"> • Lorsque la couleur noire et le clignotement sont configurés, la couleur de l'arrière-plan devient transparente. |

■ Ligne

| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Dessine une ligne à la position désignée. Désignez le type de ligne, les attributs de la couleur, et les coordonnées de début et de fin. |
| Format | <p>dsp_line (Coordonnée en X du point de départ, Coordonnée en Y du point de départ, Coordonnée en X du point final, Coordonnée en Y du point final, Clignotement de la couleur d'affichage + Couleur d'affichage, Clignotement de la couleur de l'arrière-plan + Couleur de l'arrière-plan, Type de ligne et Flèche)</p>  <p>REMARQUE</p> <ul style="list-style-type: none"> • Lorsque la couleur noire et le clignotement sont configurés, la couleur de l'arrière-plan devient transparente. |

■ **Rectangle**

| Elément | Description |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Dessine un rectangle à la position désignée. Lorsque vous cochez la case [Motif], un rectangle rempli est dessiné. Sélectionnez et saisissez le type de ligne (ou le motif du remplissage lorsque vous sélectionnez un motif), les attributs de la couleur, et les coordonnées de début et de fin.</p> |
| Format | <p>dsp_rectangle (Coordonnée en X du point de départ, Coordonnée en Y du point de départ, Coordonnée en X du point final, Coordonnée en Y du point final, Clignotement de la couleur d'affichage + Couleur d'affichage, Clignotement de la couleur de l'arrière-plan + Couleur de l'arrière-plan, Motif et Type de ligne)</p> <div data-bbox="537 602 1103 906" style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> </div> <p>REMARQUE</p> <ul style="list-style-type: none"> • Lorsque la couleur noire et le clignotement sont configurés, la couleur de l'arrière-plan devient transparente. |

IMPORTANT

- Lorsque vous utilisez des couleurs dans les fonctions de dessin, définissez des codes de couleur de 0 à 255. Si vous définissez E1 à E12 et que vous enregistrez le script, une erreur se produit.

21.11.3 Opération de mémoire

| Opération de mémoire | Résumé de la fonction |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>Adresse de décalage  « ■ Adresse de décalage » (page 21-80) Configure un décalage d'adresse.</p> |
| | <p>Comparer la mémoire  « ■ Comparer la mémoire » (page 21-82) Compare deux blocs de données aux positions spécifiées (décalage), et écrit le résultat de comparaison dans l'adresse de stockage.</p> |
| | <p>Copie de mémoire  « ■ Copie de mémoire » (page 21-84) Copie la mémoire du périphérique dans une opération.</p> |
|  | <p>Copie de mémoire (spécification de la variable)  « ■ Copie de mémoire (Variable) » (page 21-88) Copie la mémoire du périphérique dans une opération. L'adresse (copier à partir) source, l'adresse (copier vers) cible, et le nombre d'adresses peuvent être modifiés.</p> |
| | <p>Memory Ring  « ■ Memory Ring » (page 21-90) Déplace les données en cercle dans la mémoire selon le nombre désigné de blocs de mots.</p> |
| | <p>Rechercher la mémoire  « ■ Rechercher la mémoire » (page 21-93) Effectue une recherche de données en unités de bloc, et retourne (enregistre) le résultat de la recherche dans l'adresse de stockage spécifiée.</p> |
| | <p>Initialiser la mémoire  « ■ Initialiser la mémoire » (page 21-96) Initialise tous les périphériques en même temps.</p> |
| | <p>Initialiser la mémoire (spécification de la variable)  « ■ Initialiser la mémoire (variable) » (page 21-97) Initialise tous les périphériques en même temps. La première adresse, les données configurées, et le nombre d'adresses peuvent être modifiés.</p> |
| | <p>Décaler la mémoire  « ■ Décaler la mémoire » (page 21-98) Déplace les unités de bloc vers le haut.</p> |

■ Adresse de décalage

| Elément | Description | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------------------|--|-----|-----|-------|---|-------|-------|---|------------|----------|--------|-------|----------|-------------|------------|-------|---|------|-------|---|----------|
| Résumé | Vous pouvez désigner des adresses de décalage. Seules les adresses de mot temporaires peuvent être désignées pour les adresses de stockage de la valeur de décalage. | | | | | | | | | | | | | | | | | | | | | | | |
| Format | <p data-bbox="385 338 978 367">[Adresse de périphérique] # [Adresse de décalage]</p> <div data-bbox="509 382 1129 672" style="border: 1px solid gray; padding: 5px;"> </div> <p data-bbox="385 685 692 714">Plages d'entrée constantes</p> <table border="1" data-bbox="395 743 959 1058"> <thead> <tr> <th rowspan="2">Type de données</th> <th colspan="2">Saisie constante</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> | Type de données | Saisie constante | | Min | Max | Bin16 | 0 | 65535 | Bin32 | 0 | 4294967295 | Bin16+/- | -32768 | 32767 | Bin32+/- | -2147483648 | 2147483647 | BCD16 | 0 | 9999 | BCD32 | 0 | 99999999 |
| Type de données | Saisie constante | | | | | | | | | | | | | | | | | | | | | | | |
| | Min | Max | | | | | | | | | | | | | | | | | | | | | | |
| Bin16 | 0 | 65535 | | | | | | | | | | | | | | | | | | | | | | |
| Bin32 | 0 | 4294967295 | | | | | | | | | | | | | | | | | | | | | | |
| Bin16+/- | -32768 | 32767 | | | | | | | | | | | | | | | | | | | | | | |
| Bin32+/- | -2147483648 | 2147483647 | | | | | | | | | | | | | | | | | | | | | | |
| BCD16 | 0 | 9999 | | | | | | | | | | | | | | | | | | | | | | |
| BCD32 | 0 | 99999999 | | | | | | | | | | | | | | | | | | | | | | |

Exemple d'expression 1 :

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

Dans l'exemple ci-dessus, lorsque la valeur de [t:0000] est de 2, la valeur stockée dans D0102 est décalée à D0200.

Exemple d'expression 2 :

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

Dans l'exemple ci-dessus, lorsque la valeur de [t:0000] est de 8, la valeur 30 est décalée à D0108.

IMPORTANT

- Les adresses de mot utilisées en format d'adresse de décalage ne sont pas comptées en tant qu'adresses D-Script.
 - Les données provenant d'un périphérique désigné par une adresse de décalage ne sont pas lues de façon continue à partir du périphérique connecté. Elles sont lues lorsque le D-Script s'exécute. Lorsqu'une erreur se produit lors de la lecture, la valeur de la lecture est traitée comme «0». De plus, le bit 12 du relais spécial interne de l'afficheur LS2032 s'active. Lorsque la lecture des données est terminée de façon normale, le bit 12 est désactivé.
 - Si le résultat du décalage d'adresse dépasse 16 bits (valeur maximale : 65535), les bits 0 à 15 sont valides et les bits 16 et supérieurs sont rejetés.
 - Lorsque vous définissez une variable en tant que l'adresse, précisez une variable de tableau entier. Assurez-vous que le tableau entier est assez grand pour héberger toutes les adresses consécutives. Les opérations seront non valides si le tableau n'est pas assez grand pour stocker les adresses consécutives. Les opérations seront également non valides si la variable entière n'est pas un tableau.
-

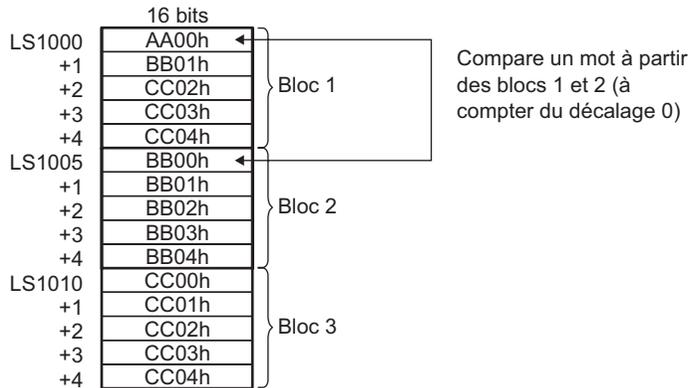
■ Comparer la mémoire

| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Compare deux blocs de données aux positions spécifiées (décalage), et écrit le résultat de comparaison dans l'adresse de stockage.</p> <p>Les valeurs suivantes sont stockées en tant que résultat de la comparaison : Lorsque les valeurs sont égales : Lorsque les données cibles sont plus nombreuses que les données originales : Lorsque les données cibles sont moins nombreuses que les données originales : Lorsqu'une erreur se produit, la valeur du statut d'erreur est écrite vers LS9152.</p> |
| Format | <p><code>_memcmp ([Adresse de bloc comparée], [Comparer à l'adresse de bloc], [Adresse de stockage du résultat de comparaison], Décaler à compter du premier bloc, Nombre de mots comparés, Nombre de mots dans 1 bloc)</code></p> <div data-bbox="491 610 1146 1016" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 : Variable interne Paramètre 2 : Variable interne Paramètre 3 : Variable interne Paramètre 4 : Valeur numérique (de 0 à 639), Variable interne, Variable temporaire Paramètre 5 : Valeur numérique (de 1 à 640) Paramètre 6 : Valeur numérique (de 1 à 640)</p> <p>Données à stocker 0: Correspondance 1: Comparer depuis < Comparer à 2: Comparer depuis > Comparer à</p> |

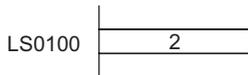
Exemple d'expression 1 :

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

(Compare un mot depuis les blocs 1 et 2 (à compter du décalage 0) et enregistre le résultat de comparaison dans LS0100)



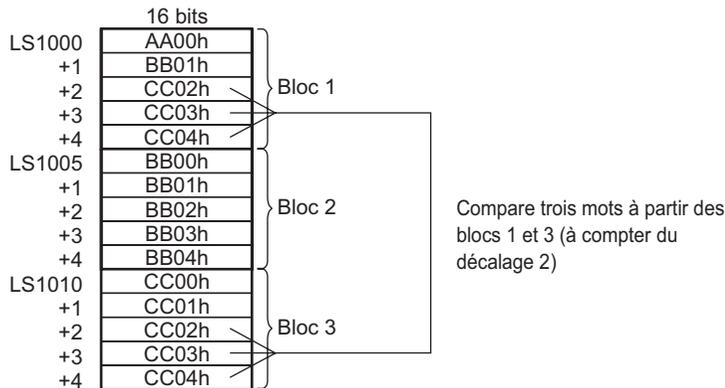
Puisque la valeur source est moins élevée que la valeur cible, le résultat de comparaison «2» est stocké dans LS0100.



Exemple d'expression 2 :

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

(Compare un mot à partir du bloc 1 et du bloc 3 (à compter du décalage 2) et enregistre le résultat de comparaison dans LS0100).



Puisque les valeurs des données originales et des données cibles correspondent, le résultat de comparaison «0» est stocké dans LS0100.



Statut d'erreur

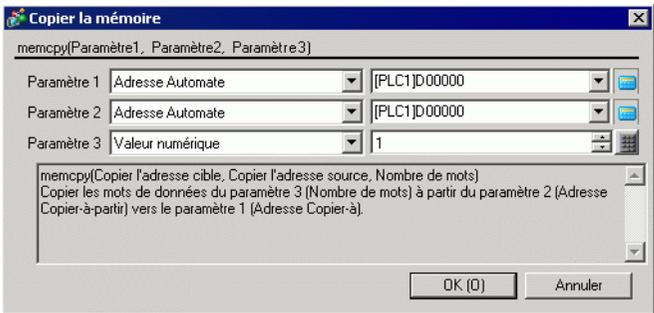
| | |
|--------|---------|
| | Zone LS |
| LS9152 | |
| | |

| Nom de fonction de l'éditeur | Zone LS | Statut d'erreur | Cause |
|------------------------------|---------|-----------------|------------------------------|
| _memcmp () | LS9152 | 0000h | Terminé avec succès |
| | | 0001h | Erreur de paramètre |
| | | 0003h | Erreur d'écriture/de lecture |

IMPORTANT

- La plage de périphérique LS efficace qui peut être spécifiée est limitée à la zone d'utilisateur désignée (de LS20 à LS2031 et de LS2096 à LS8191).
- Lorsque vous précisez une valeur qui dépasse le nombre maximum de mots dans un bloc, cette fonction ne fonctionne pas.
- Lorsque le nombre de mots à comparer dépasse la limite d'un bloc, cette fonction ne fonctionne pas.

■ Copie de mémoire

| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Copie la mémoire du périphérique dans une opération. Les données pour le nombre d'adresses sont copiées vers les adresses de mot cible à compter de la première adresse de mot des données sources. Vous pouvez utiliser de 1 à 640 adresses. |
| Format | <p>memcpy ([Adresse Copier vers], [Adresse Copier à partir], Mots)</p>  |

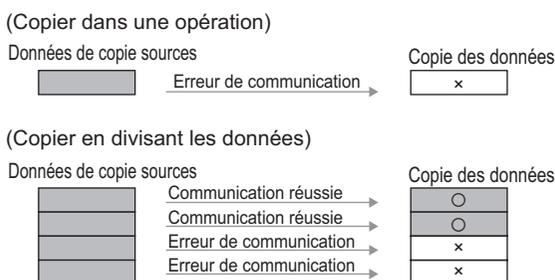
Exemple d'expression :

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

Dans l'exemple ci-dessus, les données sont copiées de D0100-D0109 à D0200-D0209.

IMPORTANT

- Les données sources sont lues à partir du périphérique connecté une seule fois, au besoin. Si une erreur de communication se produit lors de la lecture des données, le bit 12 du relais spécial interne LS2032 de l'afficheur est activé. Lorsque la lecture des données est terminée de façon normale, le bit 12 est désactivé.
- La lecture des données de copie source et l'écriture des données vers la destination s'effectuent en une seule opération, ou encore en divisant les données en plusieurs groupes équivalents au nombre d'adresses utilisées pour les données de copie sources. S'il se produit une erreur de communication pendant la lecture des données, le résultat de la copie des données variera de la façon suivante, selon si les données ont été traitées en une opération ou en plusieurs groupes. (Résultat de l'opération d'écriture de données O: Ecriture terminée, X: Impossible d'écrire)

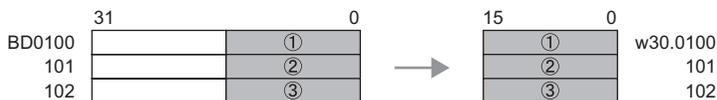


- Lorsque le nombre d'adresses augmente, il faut plus de temps pour écrire les données dans l'automate. Selon le nombre d'adresses, il peut falloir de 20 secondes à plusieurs minutes.
- Si les données à écrire dépassent la plage de périphérique désignée, une erreur de communication se produit. Dans ce cas, vous devez mettre le GP HORS tension, puis le mettre SOUS tension afin de réinitialiser le GP.
- Lorsque les données sont écrites dans la zone LS à l'aide de la fonction Copie de mémoire (memcpy), les données ne sont écrites que dans la zone utilisateur. Les données ne peuvent pas être écrites dans la zone système (de LS0000 à LS0019), dans la zone spéciale (de LS2032 à LS2047), ou dans la zone réservée (de LS2048 à LS2095). Toutefois, vous pouvez lire les données depuis ces zones.

Suite

IMPORTANT

- Lorsque vous utilisez un D-script pour copier les données de périphérique 32 bits vers un périphérique 16 bits, et que la longueur de bit est de 16 bits, seules les données dans les derniers 16 bits seront copiées.
Par exemple, memcpy ([w:[PLC1]w30.0100], [w:[PLC1]BD0100], 3)



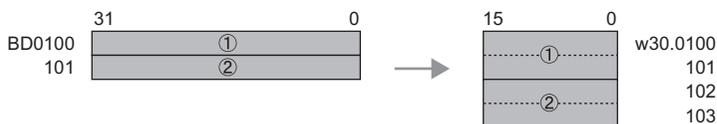
De plus, lorsque les données de périphérique 16 bits sont copiées dans un périphérique 32 bits, les données sont copiées dans les 16 bits inférieurs et «0» est défini pour les 16 bits supérieurs.

Par exemple, memcpy ([w:[PLC1]BD0100], [w:[PLC1]w30.0100], 3)

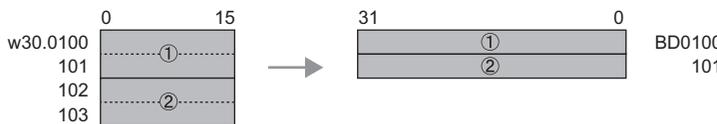


- Lorsque les données d'un périphérique 32 bits sont copiées dans un périphérique 16 bits, ou lorsque les données d'un périphérique 16 bits sont copiées dans un périphérique 32 bits, et que la longueur de bit dans le script est de 32 bits, l'opération de copie fonctionne de la façon suivante. Lorsque l'un des périphériques est un périphérique 32 bits et que l'autre est un périphérique 16 bits, la fonction memcpy () utilise les 16 bits en tant que paramètre de longueur des données.

Par exemple, memcpy ([w:[PLC1]w30.0100], [w:[PLC1]BD0100], 4)



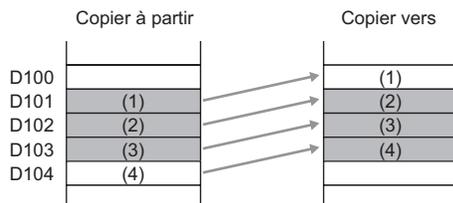
Par exemple, memcpy ([w:[PLC1]BD0100], [w:[PLC1]w30.0100], 4)



- Si les plages de données originales et cibles sont superposées, toutes les données superposées seront réécrites comme suit :

Par exemple, lors de la copie de D101-D104 vers D100-D103

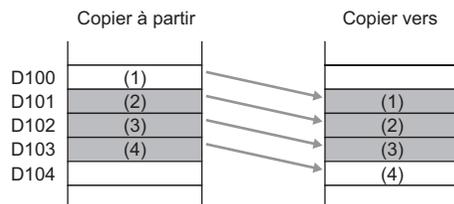
Les données sont copiées vers une adresse dont le numéro est plus petit.



Suite

IMPORTANT

Par exemple, lors de la copie de D100-D103 vers D101-D104
Les données sont copiées vers une adresse dont le numéro est plus grand.



- Bien que la fonction de cet exemple désigne 2 adresses, ces adresses ne seront pas comptées en tant qu'adresses D-Script.
 - Lorsque vous utilisez une adresse de périphérique pour l'attribution, la communication avec le périphérique/l'automate entraîne un léger retard dans l'attribution de la valeur.
-

■ Copie de mémoire (Variable)

| Élément | Description |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Copie la mémoire du périphérique dans une opération. Les données des adresses spécifiées dans le paramètre 3 sont copiées à partir de l'adresse de mot source spécifiée dans le paramètre 2 vers l'adresse de mot cible spécifiée dans le paramètre 1.</p> <p>Vous pouvez utiliser de 1 à 640 adresses et désigner indirectement l'adresse source, l'adresse cible, et le nombre d'adresses à l'aide de la fonction «_memcpy_EX».</p> |
| Format | <p>_memcpy_EX ([Adresse Copier vers], [Adresse Copier à partir], Nombre de mots)</p> <p>Paramètre 1 : Adresse de périphérique + Adresse temporaire Paramètre 2 : Adresse de périphérique + Adresse temporaire Paramètre 3 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 3 est entre 1 et 640.)</p> <div data-bbox="491 683 1146 996" style="border: 1px solid gray; padding: 5px; margin: 10px auto; width: fit-content;"> <p>Copier la mémoire (spéc de la variable)</p> <p>_memcpy_EX(Paramètre1, Paramètre2, Paramètre3)</p> <p>Paramètre 1 Périphérique avec spécification de dé [PLC1]D00000 # 0000</p> <p>Paramètre 2 Périphérique avec spécification de dé [PLC1]D00000 # 0000</p> <p>Paramètre 3 Mémoire interne [#INTERNAL]LS0000</p> <p>_memcpy_EX\Copier l'adresse cible, Copier l'adresse source, Nombre de mots</p> <p>Copier les mots de données du paramètre 3 (Nombre de mots) à partir du paramètre 2 (Adresse Copier-à-partir) vers le paramètre 1 (Adresse Copier-à).</p> <p>OK [O] Annuler</p> </div> |

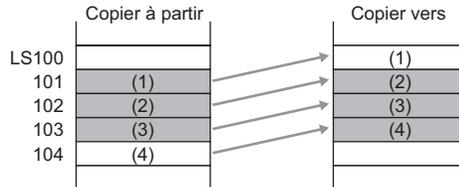
Exemple d'expression :

[t:0000]=10, [t:0001]=20
 _memcpy_EX ([w:[#INTERNAL]LS0100]#[t:0000], [w:[PLC1]D0100]#[t:0001], 5)

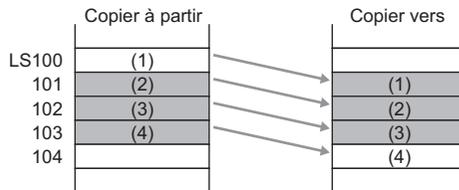
Dans l'exemple ci-dessus, cinq mots de données sont lus depuis D0120 et écrits dans LS0110 à LS0114.

IMPORTANT

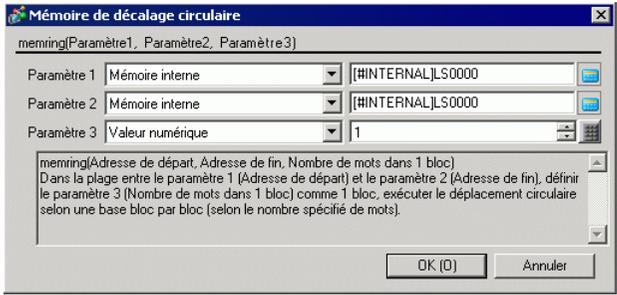
- Si les plages de données originales et cibles sont superposées, toutes les données superposées seront réécrites comme suit :
Par exemple, lors de la copie de LS101-LS104 vers LS100-LS103
Les données sont copiées vers une adresse dont le numéro est plus petit.



Par exemple, lors de la copie de LS100-LS103 vers LS101-LS104
Les données sont copiées vers une adresse dont le numéro est plus grand.

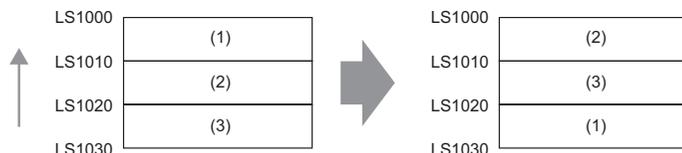


■ Memory Ring

| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Déplace les données en cercle dans la mémoire, par blocs. Effectue le déplacement en cercle entre les adresses de départ et de fin des unités de bloc (selon le nombre précisé de mots). Lorsqu'une erreur se produit, le statut de l'erreur est écrit vers LS9150. |
| Format | <p>memring ([Adresse de départ], [Adresse de fin], Nombre de mots dans 1 bloc)</p>  <p>Paramètre 1 : Variable interne Paramètre 2 : Variable interne Paramètre 3 : Valeur numérique (de 1 à 640)</p> <ul style="list-style-type: none"> - Lorsque le paramètre 1 est moins élevé que le paramètre 2 ($P1 < P2$), les données de bloc sont déplacées vers le haut. - Lorsque le paramètre 1 est plus élevé que le paramètre 2 ($P1 > P2$), les données de bloc sont déplacées vers le bas. <p>IMPORTANT</p> <ul style="list-style-type: none"> • Assurez-vous que l'adresse de départ et l'adresse de fin sont configurées au même type de périphérique (LS ou USR). |

Exemple d'expression 1 :

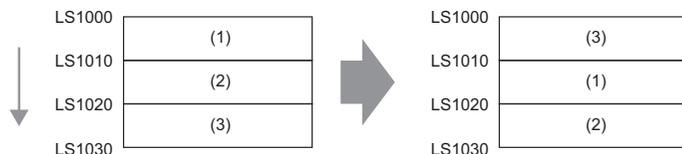
memring ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1030], 10)
 (Lorsque le paramètre 1 est plus élevé que le paramètre 2 ($P1 < P2$))



Les données sont déplacées vers le haut en unités de bloc de 10 mots.

Exemple d'expression 2 :

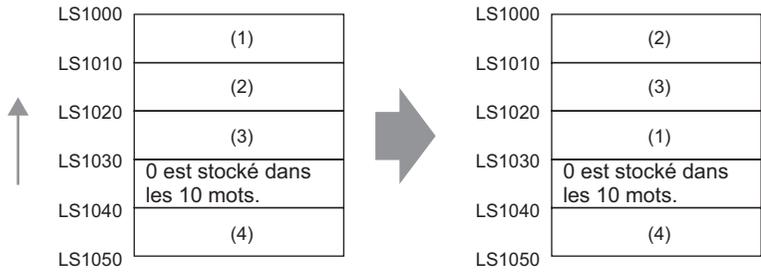
memring ([w:[#INTERNAL]LS1030], [w:[#INTERNAL]LS1000], 10)
 (Lorsque le paramètre 1 est plus élevé que le paramètre 2 ($P1 > P2$))



Les données sont déplacées vers le bas en unités de bloc de 10 mots.

Exemple d'expression 3 :

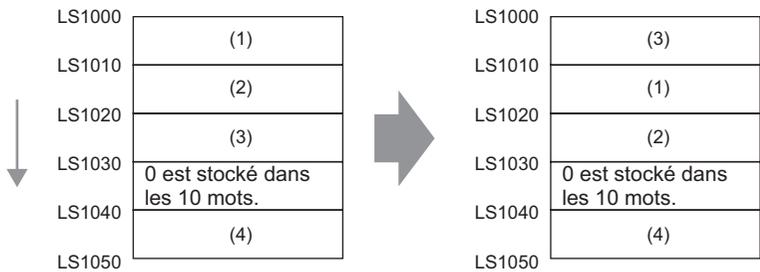
memring ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1050], 10)
 (Lorsque la plage contient un bloc pour lequel tous les mots sont «0».)



Les données se déplacent vers le haut, en unités de bloc de 10 mots, depuis le bloc de départ jusqu'au bloc qui contient les données «0». S'il existe des données après le bloc dont les données sont «0», les données sont ignorées.

Exemple d'expression 4 :

memring ([w:[#INTERNAL]LS1050], [w:[#INTERNAL]LS1000], 10)
 (Lorsqu'il existe un bloc dont les données sont «0» dans la plage.)



Les données se déplacent vers le bas, en unités de bloc de 10 mots, depuis le bloc de départ jusqu'au bloc qui contient les données «0». S'il existe des données après le bloc dont les données sont «0», les données sont ignorées.

Statut d'erreur

| | |
|--------|---------|
| LS9150 | Zone LS |
| | |
| | |

| Nom de fonction de l'éditeur | Zone LS | Statut d'erreur | Cause |
|------------------------------|---------|-----------------|------------------------------|
| memring () | LS9150 | 0000h | Terminé avec succès |
| | | 0001h | Erreur de paramètre |
| | | 0003h | Erreur d'écriture/de lecture |

IMPORTANT

- Le temps de traitement nécessaire est proportionnel à la plage désignée par les adresses de départ et de fin. Plus la plage désignée est grande, plus il faudra de temps pour le traitement. L'objet n'est pas actualisé tant que le traitement n'est pas terminé.
- La plage de périphérique LS efficace qui peut être spécifiée est limitée à la zone d'utilisateur désignée (de LS20 à LS2031 et de LS2096 à LS8191).

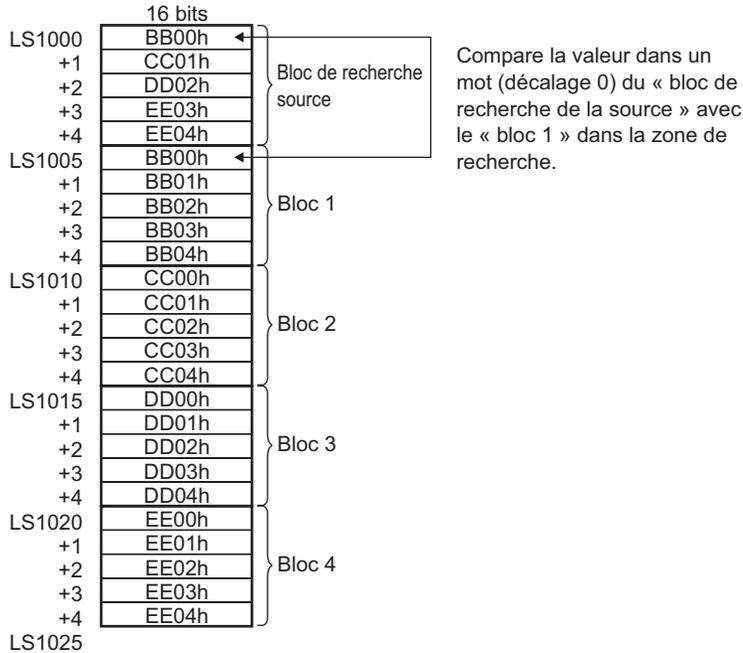
■ Rechercher la mémoire

| Elément | Description |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Effectue une recherche de données en unités de bloc, à compter du premier élément dans la plage spécifiée. Compare les blocs de données, à compter des blocs spécifiés (décalage) et retourne (enregistre) le résultat de la recherche dans l'adresse de stockage spécifiée. Lorsqu'un bloc correspondant est trouvé, la valeur de décalage du bloc (1 ou plus élevé) est enregistrée. Lorsqu'on ne trouve aucun bloc correspondant, «FFFFh» est enregistré. Lorsqu'une erreur se produit, la valeur du statut d'erreur est écrite vers LS9153.</p> |
| Format | <p>_memsearch ([Adresse de bloc recherchée], [Adresse de départ de la recherche], [Adresse de fin de la recherche], [Adresse de stockage du résultat de la recherche], Décaler à compter du premier bloc, Nombre de mots comparés, Nombre de mots dans 1 bloc)</p> <div data-bbox="491 668 1149 1103" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 : Variable interne Paramètre 2 : Variable interne Paramètre 3 : Variable interne Paramètre 4 : Variable interne Paramètre 5 : Valeur numérique (de 0 à 639), Variable interne, Variable temporaire Paramètre 6 : Valeur numérique (de 1 à 640) Paramètre 7 : Valeur numérique (de 1 à 640)</p> <p>Données à écrire Lorsqu'il y a des blocs correspondants : La valeur de décalage du bloc («1» ou plus élevée) Lorsqu'il n'y a aucun bloc correspondant : «FFFFh»</p> <div data-bbox="389 1547 522 1591" style="border: 1px solid black; padding: 2px; margin: 10px 0;"> <p>IMPORTANT</p> </div> <ul style="list-style-type: none"> • Assurez-vous que l'adresse de départ de la recherche et l'adresse de fin de la recherche sont configurées au même type de périphérique (LS ou USB). Toutefois, l'[adresse de bloc recherchée] et l'[adresse de stockage du résultat de la recherche] peuvent être configurées à la mémoire interne. • Assurez-vous que le [Paramètre 2] est moins élevé que le [Paramètre 3]. Sinon, une erreur se produit. |

Exemple d'expression 1 :

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

(Recherche de LS1005 à LS1025 un bloc dont la valeur est la même. Commence à partir du décalage 0 du bloc de recherche source et stocke le résultat dans LS0100.)



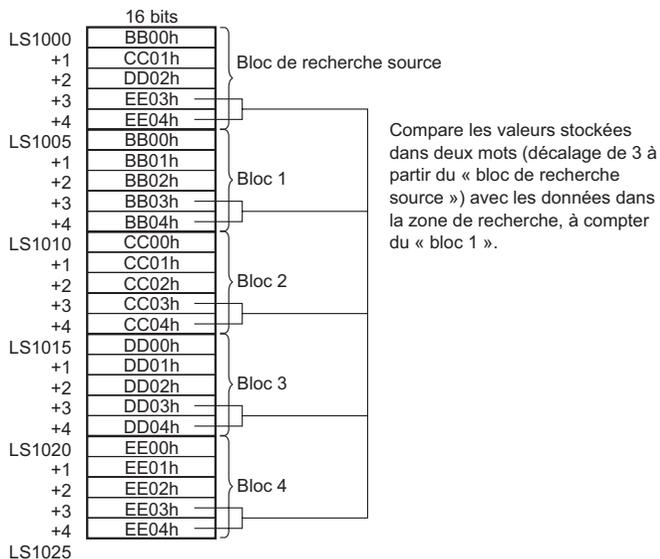
Dans ce cas, la valeur du «Bloc 1» correspond à la valeur du «bloc de recherche source». Par conséquent, le résultat de recherche «1» est stocké dans LS0100.

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

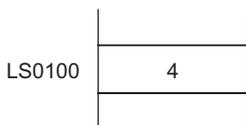
Exemple d'expression 2 :

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

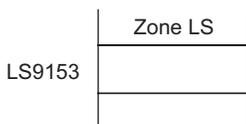
(Recherche de LS1005 à LS1025 un bloc dont la valeur est la même. Utilise deux mots, à compter d'un décalage de 3, et stocke le résultat dans LS0100.)



Dans ce cas, la valeur du «Bloc 4» correspond à la valeur du «bloc de recherche source». Par conséquent, le résultat de recherche «4» est stocké dans LS0100.



Statut d'erreur



| Nom de fonction de l'éditeur | Zone LS | Statut d'erreur | Cause |
|------------------------------|---------|-----------------|------------------------------|
| _memsearch () | LS9153 | 0000h | Terminé avec succès |
| | | 0001h | Erreur de paramètre |
| | | 0003h | Erreur d'écriture/de lecture |

IMPORTANT

- Le temps de traitement nécessaire est proportionnel à la plage désignée par les adresses de départ et de fin. Plus la plage désignée est grande, plus il faudra de temps pour le traitement. L'objet n'est pas actualisé tant que le traitement n'est pas terminé.
- La plage de périphérique LS efficace qui peut être spécifiée est limitée à la zone d'utilisateur désignée (de LS20 à LS2031 et de LS2096 à LS8191).

■ Initialiser la mémoire

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Initialise tous les périphériques en même temps. La configuration des données pour le nombre d'adresses correspond à l'adresse de mot configurée. La plage valide pour le nombre d'adresses est de 1 à 640. |
| Format | memset ([Adresse Ecrire vers], Ecrire les données, Nombre de mots)  |

Exemple d'expression :

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

Dans l'exemple ci-dessus, la valeur «0» est configurée pour les adresses comprises entre D0100 et D0109.

IMPORTANT

- Lorsque le nombre d'adresses augmente, il faut plus de temps pour écrire les données dans l'automate. Selon le nombre d'adresses, il peut falloir de 20 secondes à plusieurs minutes.
- Si les données à écrire dépassent la plage de périphérique désignée, une erreur de communication se produit. Dans ce cas, vous devez mettre le GP HORS tension, puis le mettre SOUS tension afin de réinitialiser le GP.
- Bien que cette fonction désigne des adresses, elles ne sont pas comptées en tant qu'adresses D-Script.
- Lorsque vous écrivez des données dans la zone LS à l'aide de la fonction Rétablir la mémoire (memset), les données ne peuvent être écrites que dans la zone utilisateur. Les données ne peuvent pas être écrites dans la zone système (de LS0000 à LS0019), dans la zone spéciale (de LS2032 à LS2047), ou dans la zone réservée (de LS2048 à LS2095).
- Lorsque vous utilisez des adresses de périphérique pour l'opération d'affectation, les valeurs d'écriture ne sont pas affectées immédiatement, en raison de la durée de transmission GP à automate.

Par exemple :

```
memset ([w:[PLC1]D0100], 0, 10) //Initialiser D100-D109 sur 0
[w:[PLC1]D200]=[w:[PLC1]D100] //Substituer D100 à D200
```

Dans ce cas, le résultat de l'opération écrit dans D100 n'est toujours pas assigné à D200.

■ Initialiser la mémoire (variable)

| Elément | Description |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Initialise tous les périphériques en même temps. Les données spécifiées dans le paramètre 2 sont configurées selon l'adresse de mot spécifiée dans le paramètre 1 vers les adresses spécifiées dans le paramètre 3. La plage valide pour le nombre d'adresses est de 1 à 640. L'adresse Ecrire vers, les données d'écriture, et le nombre d'adresses peuvent être désignés de façon indirecte. |
| Format | <p data-bbox="385 455 1243 484"><code>_memset_EX ([Adresse Ecrire vers], Ecrire les données, Nombre de mots)</code></p> <div data-bbox="491 504 1146 813" style="border: 1px solid gray; padding: 5px;"> </div> <p data-bbox="385 832 1243 1010"> Paramètre 1 : Adresse de périphérique + Adresse temporaire Paramètre 2 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 2 est de 0 à 65535 pour Déc., et de 0 à FFFF pour Hex.) Paramètre 3 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 3 est entre 1 et 640.) </p> |

Exemple d'expression :

`[t:0000]=10`

`[w:[#INTERNAL]LS0050]=0`

`[w:[#INTERNAL]LS0051]=5`

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

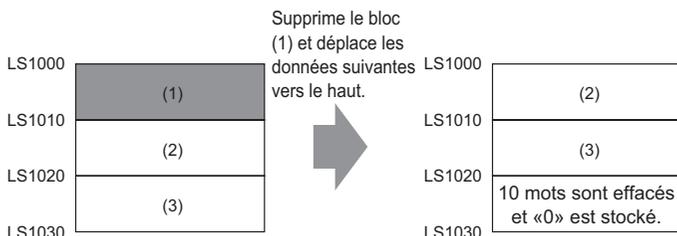
Dans l'exemple ci-dessus, la valeur «0» sera écrite dans les cinq mots provenant de LS0100 à LS0114.

■ Décaler la mémoire

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Supprime le bloc spécifié et déplace les blocs de données suivants vers le haut. Le bloc à supprimer est désigné à l'aide d'un décalage. Lorsqu'une erreur se produit, le statut de l'erreur est écrit vers LS9151. |
| Format | <p>_memshift ([Adresse de départ], [Adresse de fin], Décalage du bloc à supprimer, Nombre de mots dans 1 bloc)</p> <div data-bbox="495 440 1146 780" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 : Variable interne Paramètre 2 : Variable interne Paramètre 3 : Valeur numérique (de 1 à 65 535), Variable interne, Variable temporaire Paramètre 4 : Valeur numérique (de 1 à 640)</p> <div data-bbox="388 967 522 1006" style="border: 1px solid black; padding: 2px; margin: 10px 0;">IMPORTANT</div> <ul style="list-style-type: none"> • Assurez-vous que l'adresse de départ et l'adresse de fin sont configurées au même type de périphérique (LS ou USR). • Assurez-vous que le [Paramètre 1] est moins élevé que le [Paramètre 2]. Sinon, une erreur se produit. |

Exemple d'expression 1 :

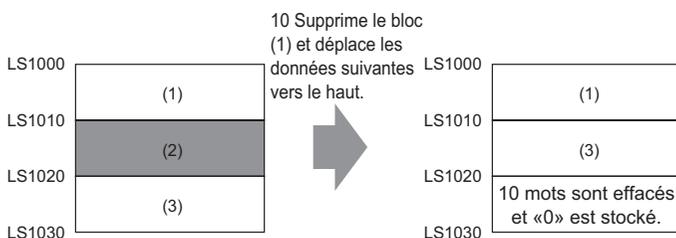
`_memshift ([w:[#INTERNAL]LS 1000], [w:[#INTERNAL]LS1030], 1, 10)`



Les données sont déplacées vers le haut en unités de bloc (1 bloc = 10 mots), et le dernier bloc (10 mots) est rétabli à zéro.

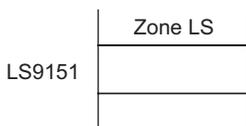
Exemple d'expression 2 :

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



Les données se déplacent vers le haut en unités de bloc (1 bloc = 10 mots), à compter de la position de décalage 2, et le dernier bloc (10 mots) est remis à zéro.

Statut d'erreur



| Nom de fonction de l'éditeur | Zone LS | Statut d'erreur | Cause |
|------------------------------|---------|-----------------|------------------------------|
| _memshift () | LS9151 | 0000h | Terminé avec succès |
| | | 0001h | Erreur de paramètre |
| | | 0003h | Erreur d'écriture/de lecture |

IMPORTANT

- Le temps de traitement nécessaire est proportionnel à la plage désignée par les adresses de départ et de fin. Plus la plage désignée est grande, plus il faudra de temps pour le traitement. L'objet n'est pas actualisé tant que le traitement n'est pas terminé.
- Lorsqu'une valeur qui dépasse la plage spécifiée pour les adresses de départ et de fin est désignée en tant que décalage du bloc à supprimer, cette fonction ne s'exécute pas correctement.
- La plage de périphérique LS efficace qui peut être spécifiée est limitée à la zone d'utilisateur désignée (de LS20 à LS2031 et de LS2096 à LS8191).

21.11.4 Opération de port SIO

| Opération de port SIO | Résumé de la fonction |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>The screenshot shows a window titled 'Fonction' with a dropdown menu 'Fonction intégrée (instruction)' set to 'Opération de port SIO'. Below the dropdown is a list of options: 'Etiquette', 'Recevoir', 'Envoyer', 'Mise en veille', and 'Réception en attente'. An 'Entrée' button is at the bottom.</p> | <p>Paramètres d'étiquette ☞ « ■ Paramètres d'étiquette » (page 21-102) Configurés selon le contrôle, le statut, le nombre de données reçues, la fonction de réception, et la fonction d'envoi.</p> |
| | <p>Recevoir ☞ « ■ Recevoir » (page 21-104) Lit les données reçues à partir du port série désigné (COM1 ou COM2).</p> |
| | <p>Envoyer ☞ « ■ Envoyer » (page 21-105) Ecrit vers le port série désigné (COM1 ou COM2).</p> |
| | <p>Réception étendue ☞ « ■ Réception étendue » (page 21-106) Lit les données reçues à partir du port série désigné (COM1 ou COM2). Ne peut être utilisé que dans un script étendu.</p> |
| | <p>Envoi étendu ☞ « ■ Envoi étendu » (page 21-107) Ecrit vers le port série désigné (COM1 ou COM2). Ne peut être utilisé que dans un script étendu.</p> |
| | <p>Fonction de réception en veille ☞ « ■ Fonction de réception de mise en veille » (page 21-108) Demeure en mode de réception en veille jusqu'à ce que le texte spécifié soit reçu. Ne peut être utilisé que dans un script étendu.</p> |
| | <p>Fonction de mise en veille ☞ « ■ Fonction de mise en veille » (page 21-109) Le système attend pendant la période précisée. Ne peut être utilisé que dans un script étendu.</p> |

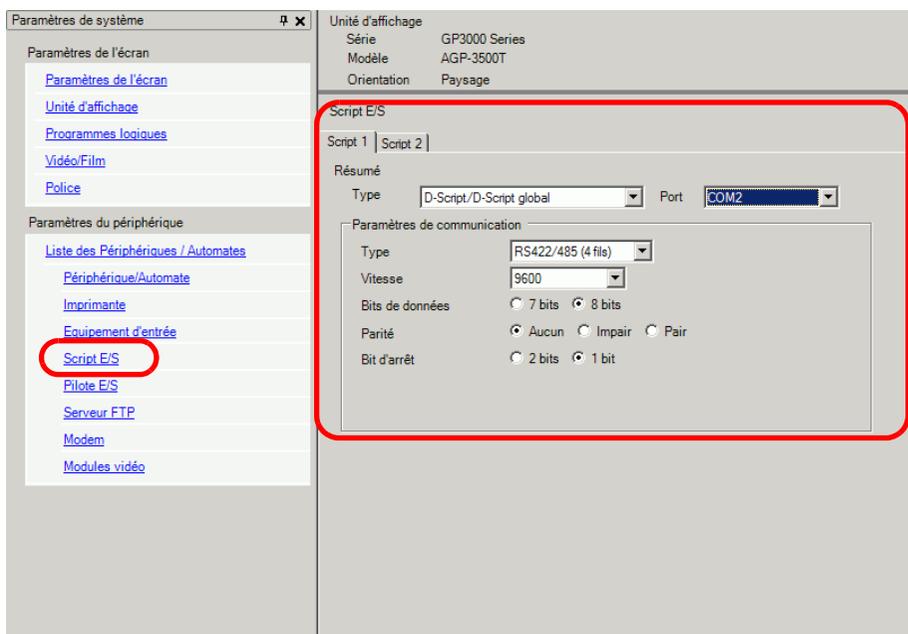
IMPORTANT

- Vous pouvez inclure facilement des paramètres d'étiquette, d'envoi, et de réception dans un D-Script/D-Script global.
- Configurer les paramètres de script suivants afin de communiquer avec les D-Scripts/D-Scripts globaux. Si les paramètres de script ne sont pas désignés, vous ne pouvez pas les exécuter.

[Procédure E/S D-Script/D-Script global]

Cliquez sur [Script] dans la fenêtre [Paramètres système].

Définissez le [Type] sur [D-Script/D-Script global].



Il y a deux onglets pour les paramètres de script. L'exemple ci-dessus utilise [Script1].

Configurez le [Port] à COM1 ou à COM2, puis configurez les [Paramètres de communication] afin de correspondre au SIO étendu.

- Lorsque vous créez un programme de communication ayant une fonctionnalité plus développée que l'opération du port SIO, il vous est recommandé d'utiliser un [Script étendu]. Pour des exemples sur la façon d'utiliser les scripts détaillés, reportez-vous à la section «21.5 Communication avec des périphériques non pris en charge» (page 21-21).

■ Paramètres d'étiquette

Contrôle

| Elément | Description |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Cette variable de contrôle sert à effacer le tampon d'envoi, le tampon de réception, et le statut d'erreur. Cette variable de contrôle est en écriture seule. |
| Format | Lorsque vous désignez le bit : [c:EXT_SIO_CTRL**]**: de 00 à 15) Lorsque vous désignez le mot : [c:EXT_SIO_CTRL] |

◆ Exemple d'expression

Lorsque vous désignez le bit : [c:EXT_SIO_CTRL00] = 1

Lorsque vous désignez le mot : [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 | Contenu |
|-----|------------------------------------------|
| 15 | Réservé |
| 14 | |
| 13 | |
| 12 | |
| 11 | |
| 10 | |
| 9 | |
| 8 | |
| 7 | |
| 6 | |
| 5 | |
| 4 | |
| 3 | 1: Effacer la temporisation de réception |
| 2 | 1: Erreur d'effacement |
| 1 | 1: Effacer le tampon de réception |
| 0 | 1: Effacer le tampon d'envoi |

REMARQUE

- Lorsqu'un mot est sélectionné et que deux bits ou plus sont configurés simultanément, le traitement est exécuté dans l'ordre suivant : Effacer l'erreur -> Effacer le tampon de réception -> Effacer le tampon d'envoi

Statut

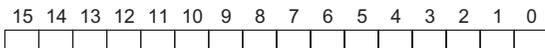
| Elément | Description |
|---------|------------------------------------------------------------------------------------------------------------------------|
| Résumé | Le statut inclut les informations suivantes : Cette variable de statut est en écriture seule. |
| Format | Lorsque vous désignez le bit : [s:EXT_SIO_STAT**] (** : de 00 à 15) Lorsque vous désignez le mot : [s:EXT_SIO_STAT] |

◆ Exemple d'expression

Lorsque vous désignez le bit : if ([s:EXT_SIO_STAT 00] == 1)

Lorsque vous désignez le mot : if (([s:EXT_SIO_STAT] & 0x0001) <> 0)

◆ **Contenu de EXT_SIO_STAT**



| Bit | Contenu |
|-----|---------------------------------------------------------------------------------|
| 15 | 0: Aucun D-Script/D-Script global 1: Un D-Script/D-Script global existe |
| 14 | 0: Aucun script étendu 1: Script étendu existant |
| 13 | Réservé |
| 12 | |
| 11 | |
| 10 | |
| 9 | |
| 8 | |
| 7 | |
| 6 | |
| 5 | 0: Normal 1: Temporisation de réception |
| 4 | |
| 3 | 0: Normal 1: Erreur de réception |
| 2 | 0: Aucune donnée de réception 1: Données de réception existantes |
| 1 | 0: Normal 1: Erreur d'envoi |
| 0 | 0: Des données existent dans le tampon Envoyer 1: Le tampon d'envoi est vide |

REMARQUE

- Les bits réservés peuvent être affectés ultérieurement. Donc, assurez-vous de ne vérifier que les bits nécessaires.
- Il existe deux types d'erreur de transmission : l'erreur de temporisation de transmission et l'erreur de tampon complet de transmission. Lorsqu'une des deux erreurs se produit, le bit d'erreur de transmission s'activera. La période de temporisation de la transmission est de cinq secondes.
- Il existe quatre types d'erreur de réception : l'erreur de parité, l'erreur de dépassement, l'erreur de cadrage, et l'erreur de débordement. Lorsque l'une de ces erreurs se produit, le bit pour l'erreur de réception est activé.
- Si une erreur de transmission est détectée, les données d'envoi demeurent dans le tampon de transmission. S'il est impossible de détecter une erreur de transmission, les données d'envoi sont envoyées à partir du tampon de transmission.
- Lorsque vous utilisez l'interface série COM2, en RS-422, il est impossible de détecter le signal CS (CTS). Par conséquent, il est impossible de détecter le débranchement d'un câble.

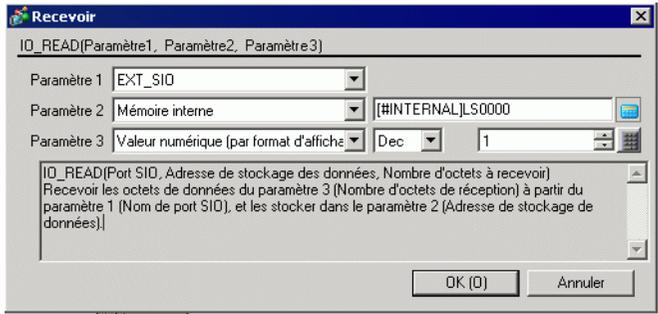
Taille des données reçues

| Élément | Description |
|---------|--------------------------------------------------------------------------------------------------------------------------|
| Résumé | Indique la quantité des données (nombre d'octets) reçues à ce moment. La taille des données reçues est en lecture seule. |
| Format | [r:EXT_SIO_RECV] |

IMPORTANT

- Nom d'étiquette du nombre de données reçues (nombre d'octets)
Avec GP-PRO/PB III V. 6.0 et les versions plus anciennes, le nom de l'étiquette désignée pour la taille des données reçues est [r: EXT_SIO_RCV]. Cependant, vous n'êtes pas tenu de réviser la description, puisque la fonction est la même que l'expression [r: EXT_SIO_RCV] ou [r: EXT_SIO_RECV].

■ **Recevoir**

| Elément | Description |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Ecrivez l'instruction comme suit lors de la lecture des données reçues à partir du SIO étendu. |
| Format | <p>IO_READ ([p:EXT_SIO], Adresse de stockage des données, Nombre d'octets reçus)</p>  <p>Paramètre 1 : EXT_SIO Paramètre 2 : Variable interne Paramètre 3 : Valeur numérique</p> |

Exemple d'expression :

IO_READ ([p:EXT_SIO], [w:[#INTERNAL]LS0100], 10)

Dans l'exemple ci-dessus, le nombre d'octets reçus est stocké dans LS0100. 10 octets de données sont stockés à compter de LS0101. L'image suivante montre les données reçues stockées.

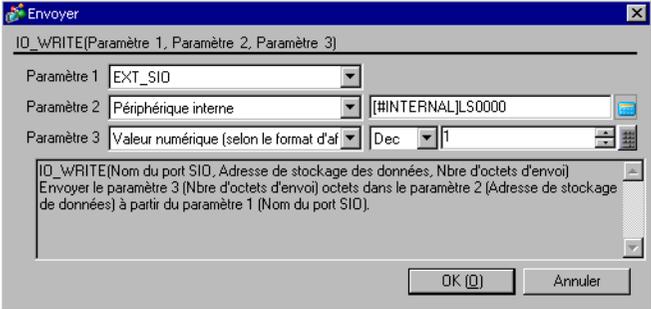
REMARQUE

- Le nombre maximum d'octets de transfert lors de la réception des données est de 2011. Les données sont écrites dans chaque adresse de mot en 1 octet.

| Adresse | Taille de données reçues | Contenu |
|---------|--------------------------|---------------|
| LS0100 | | ... 10 octets |
| LS0101 | 00 | Octet 1 |
| LS0102 | 00 | Octet 2 |
| LS0103 | 00 | Octet 3 |
| LS0104 | 00 | Octet 4 |
| LS0105 | 00 | Octet 5 |
| LS0106 | 00 | Octet 6 |
| LS0107 | 00 | Octet 7 |
| LS0108 | 00 | Octet 8 |
| LS0109 | 00 | Octet 9 |
| LS0110 | 00 | Octet 10 |

Méthode de stockage des données reçues

■ Envoyer

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Ecrivez l'instruction comme suit lors de l'écriture des données vers le SIO étendu. |
| Format | <p>IO_WRITE ([p:EXT_SIO], Adresse de stockage des données, Nombre d'octets envoyés)</p>  <p>Paramètre 1 : EXT_SIO Paramètre 2 : Variable interne Paramètre 3 : Valeur numérique</p> |

Exemple d'expression :

IO_WRITE ([p:EXT_SIO], [w:[#INTERNAL]LS0100], 10)

Dans l'exemple ci-dessus, 10 octets de données à compter de LS0100 sont envoyés. L'image suivante montre les données envoyées stockées.

REMARQUE

- Le nombre maximum d'octets de transfert lors de la réception des données est de 2012.
- Le périphérique LS envoie un octet par mot dans le tampon de données.

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

Méthode de stockage des données envoyées

■ Réception étendue

| Elément | Description |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Reçoit les données de la taille indiquée dans Taille des données reçues (octets) depuis SIO étendu et les stocke dans le tampon de données. Le nombre d'octets précisé avec le Paramètre 3 est reçu de SIO étendu et stocké dans le tampon des données précisé avec le Paramètre 2. Ne peut être utilisé que dans un script étendu. |
| Format | <p>IO_READ_EX ([p:EXT_SIO], Tampon de données, Nombre d'octets de réception)</p> <div data-bbox="495 504 1149 815" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 : [p:EXT_SIO] Paramètre 2 : Tampon de données Paramètre 3 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 3 est entre 1 et 1024.)</p> |

Exemple d'expression :

IO_READ_EX ([p:EXT_SIO], databuf1, 10)

Dans l'exemple ci-dessus, 10 octets de données dans les données reçues par le SIO étendu sont reçus et stockés dans «databuf1».

■ **Envoi étendu**

| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Envoie les données dans le tampon de données avec le SIO étendu selon la taille du nombre d'octets d'envoi. Le contenu du tampon de données spécifié dans le paramètre 2 est envoyé à partir du SIO étendu la longueur précisée dans le paramètre 3. Ne peut être utilisé que dans un script étendu.</p> |
| Format | <p>IO_WRITE_EX ([p:EXT_SIO], Tampon de données, Nombre d'octets d'envoi)</p> <div data-bbox="495 465 1149 778" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 : [p:EXT_SIO] Paramètre 2 : Tampon de données Paramètre 3 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 3 est entre 1 et 1024.)</p> |

Exemple d'expression :

IO_WRITE_EX ([p:EXT_SIO], databuf0, 10)

Dans l'exemple ci-dessus, 10 octets de données dans «databuf0» sont envoyés à partir du SIO étendu.

■ **Fonction de réception de mise en veille**

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Demeure en mode de réception en veille jusqu'à ce que le texte spécifié soit reçu. Après que la période de temporisation a expiré, le bit 4 (erreur de temporisation de réception) du statut [s: EXT_SIO_STAT] est configuré. La durée de la temporisation peut être configurée en périodes de 100 ms.</p> <p>Le système est en mode veille jusqu'à ce qu'il reçoive la chaîne de caractères ou le code de caractère spécifié dans le paramètre 2. Configurez la durée de la temporisation dans le paramètre 3.</p> <p>Ne peut être utilisé que dans un script étendu.</p> |
| Format | <p>IO_READ_WAIT([p:EXT_SIO], Texte, Délai d'attente)</p> <div data-bbox="491 614 1142 923" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 : [p:EXT_IO] Paramètre 2 : Valeur numérique, Texte, Tampon de données Paramètre 3 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 3 est entre 1 et 600.)</p> |

IMPORTANT

- Les données reçues ne peuvent être utilisées avant que le texte précisé n'ait été reçu. (Sinon, les données sont abandonnées.)
- Vous pouvez spécifier jusqu'à 128 caractères (octets). Remarquez que l'opération de réception en attente ne peut être effectuée avec succès lorsque des chaînes qui dépassent les limites sont précisées.

■ **Fonction de mise en veille**

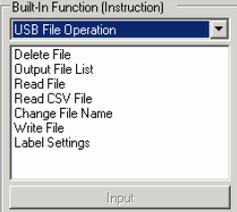
| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Le système attend pendant la période précisée. La durée peut être configurée en périodes de 100 ms. Ne peut être utilisé que dans un script étendu.</p> |
| Format | <p><code>_wait(Temps d'attente)</code></p> <div data-bbox="492 403 1146 712" style="border: 1px solid gray; padding: 5px; margin: 10px auto; width: fit-content;"> </div> <p>Paramètre 1 : Variable interne, Adresse temporaire, Valeur numérique (la plage valide pour le paramètre 1 est entre 1 et 600.)</p> |

Exemple d'expression :

`_wait (10)`

Dans l'exemple ci-dessus, le système attend une seconde.

21.11.5 Opération de fichier CF/Opération de fichier USB

| Opération de fichier CF | Résumé de la fonction |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | <p>Paramètres d'étiquette ☞ « ■ Paramètres d'étiquette » (page 21-111) Désignés selon le nombre de fichiers répertoriés, le nombre d'octets de lecture, et le statut d'erreur de la carte CF ou du stockage USB.</p> |
| | <p>Ecrire le fichier ☞ « ■ Ecrire le fichier » (page 21-122) Vous pouvez sélectionner l'un des trois modes suivants :</p> |
| | <p>Modifier le nom de fichier ☞ « ■ Modifier le nom de fichier » (page 21-127) Modifie le nom du fichier.</p> |
| | <p>Lire le fichier CSV ☞ « ■ Lire le fichier CSV » (page 21-129) Lit les données en unités de cellule à partir d'un fichier CSV et les écrit dans une adresse de mot.</p> |
| | <p>Lire le fichier ☞ « ■ Lire le fichier » (page 21-132) Lit le nombre précisé d'octets de données dans le fichier après le décalage précisé et l'écrit dans l'adresse de destination.</p> |
| | <p>Produire la liste de fichiers ☞ « ■ Produire la liste de fichiers » (page 21-135) La liste de fichiers qui existent dans le dossier spécifié est écrite dans la mémoire interne.</p> |
| | <p>Supprimer le fichier ☞ « ■ Supprimer le fichier » (page 21-137) Supprime le fichier.</p> |

■ Paramètres d'étiquette

Voici quelques valeurs de statut possibles pour le statut de carte CF/stockage USB.

| Nom de statut | Nom de l'étiquette | Description |
|---------------------------------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Fichiers répertoriés | [s:CF_FILELIST_NUM] [s:USB_FILELIST_NUM] | Stocke le nombre de fichiers actuellement énumérés lorsqu'on exécute la fonction de sortie de liste de fichiers «_CF_dir ()» ou «_USB_dir ()». |
| Nombre d'octets de lecture | [s:CF_READ_NUM] [s:USB_READ_NUM] | Stocke le nombre d'octets que l'on peut lire lorsqu'on exécute la fonction de lecture de fichier «_CF_read ()» ou «_USB_read ()». |
| Statut d'erreur de la carte CF ou du stockage USB | [s:CF_ERR_STAT] [s:USB_ERR_STAT] | Stocke le statut d'erreur généré à l'accès de la carte CF ou du stockage USB. |

Fichiers répertoriés

Lorsqu'on exécute la fonction de sortie de liste de fichiers «_CF_dir ()» ou «_USB_dir», le nombre de listes de fichier qui sont en vérité écrites dans la zone LS est stocké dans «Fichiers énumérés [s:CF_FILELIST_NUM]/[s:USB_FILELIST_NUM]».

◆ Exemple d'usage

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

```
DATA — DATA0000.BIN
      — DATA0001.BIN
      — DATA02.BIN
      — DATA0003.BIN
      — DATA0004.BIN
```

Lorsque vous obtenez une liste des 10 fichiers et que le dossier spécifié ne contient que cinq fichiers, la valeur «5» est stockée dans [s:CF_FILELIST_NUM].

IMPORTANT

- Lorsque aucun fichier n'est écrit, le nombre total de fichiers que contient le dossier précisé est écrit dans [s:CF_FILELIST_NUM].

Nombre d'octets de lecture

Lorsque la fonction de lecture de fichier «_CF_read ()» ou «_USB_read ()» est exécutée, le nombre d'octets réellement lus est stocké dans «Octets de lecture [s:CF_READ_NUM]/[s:USB_READ_NUM]».

◆ Exemple d'usage

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

Lorsque vous tentez de lire 16 octets, mais que seuls 12 octets sont lus avec succès, «12» est stocké dans [s:CF_READ_NUM].

Statut d'erreur de la carte CF ou du stockage USB

Stocke le statut d'erreur généré à l'accès de la carte CF ou du stockage USB.

| Position du bit | Nom de l'erreur | Description |
|-----------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15 | Réservé | Réservé |
| 14 | | |
| 13 | | |
| 12 | | |
| 11 | | |
| 10 | | |
| 9 | | |
| 8 | | |
| 7 | | |
| 6 | Erreur de renommage du fichier | <ul style="list-style-type: none"> • La carte CF ou le stockage USB est retiré lors de l'exécution. • Le fichier spécifié n'existe pas. |
| 5 | Erreur de suppression du fichier | <ul style="list-style-type: none"> • La carte CF ou le stockage USB est retiré lors de l'exécution. • Le fichier spécifié n'existe pas. • Il y a eu tentative de suppression d'un fichier qui possède un attribut à lecture seule. |
| 4 | Erreur d'écriture du fichier | <ul style="list-style-type: none"> • La carte CF ou le stockage USB est retiré lors de l'exécution. • Dépassement de la capacité d'espace libre dans la carte CF ou le stockage USB. • Il y a eu tentative d'écriture des données dans un fichier qui possède un attribut à lecture seule. • Une tentative «d'écrasement» a été faite pour un fichier qui n'existe pas. |
| 3 | Erreur de lecture du fichier | <ul style="list-style-type: none"> • La carte CF ou le stockage USB est retiré lors de l'exécution. • Le fichier spécifié n'existe pas. |
| 2 | Erreur de liste de fichiers | <ul style="list-style-type: none"> • La carte CF ou le stockage USB est retiré lors de l'exécution. • Le dossier spécifié n'existe pas. |
| 1 | Erreur de carte CF ou de stockage USB | <ul style="list-style-type: none"> • La carte CF ou le stockage USB est non valide. • Le média inséré n'est pas une carte CF. |
| 0 | Manque de carte CF ou de stockage USB | <ul style="list-style-type: none"> • La carte CF ou le stockage USB n'est pas inséré. • Le capot est ouvert. |

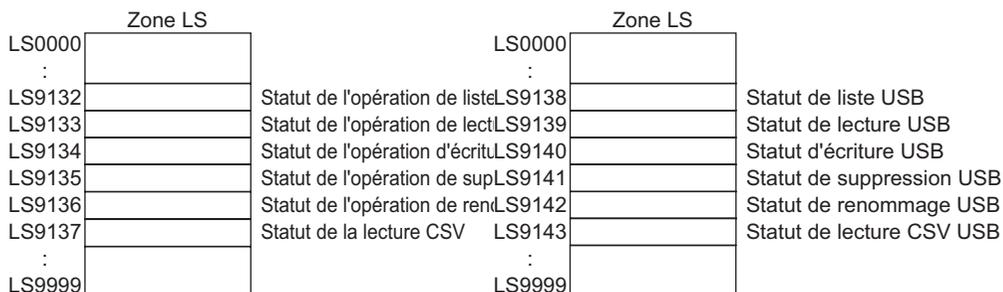
- Même si une erreur de carte CF ou de stockage USB se produit, le traitement se poursuit. Assurez-vous d'écrire le script pour qu'il vérifie s'il y a des erreurs lorsque vous utilisez les fonctions d'opération de fichier sur une carte CF ou un stockage USB.

Par exemple :

```
_CF_dir ("\\DATA\\*.*", [w:[#INTERNAL]LS0100], 2, 1) Produit une liste de fichiers.
if([s:CF_ERR_STAT02] <> 0) // Vérifie le statut d'erreur.
{
    set ([b:[#INTERNAL]LS005000]) // Définit l'adresse de bit pour l'affichage
d'erreur
}
endif
```

◆ **Zone de stockage du statut détaillé de l'erreur de carte CF ou de stockage USB**

Chaque bit est configuré lorsqu'une erreur se produit. Vous pouvez vérifier quels facteurs ont causé l'erreur en configurant le statut détaillé. Dans chaque fonction, le statut détaillé est stocké dans LS9132 à LS9137 pour la zone système (LS9138 à LS9143 pour le stockage USB). Ces zones sont uniquement en lecture seule.



◆ **Liste d'erreurs pour chaque fonction**

| Nom de fonction de l'éditeur | | Statut d'erreur | Cause |
|------------------------------|--------|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| _CF_dir () | LS9132 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier spécifié dans la chaîne fixe) |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0018H | Erreur de plage d'écriture de la zone LS |
| | | 0020h | Aucune carte CF |
| | | 0021h | Carte CF non valide |
| | | 0100h | Erreur d'ouverture du répertoire |

Suite

| Nom de fonction de l'éditeur | | Statut d'erreur | Cause |
|------------------------------|--------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| _CF_read () | LS9133 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier/de fichier spécifié dans la chaîne fixe) |
| | | 0011H | Erreur de plage de lecture de la zone LS |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0018H | Erreur de plage d'écriture de la zone LS |
| | | 0020h | Aucune carte CF |
| | | 0021h | Carte CF non valide |
| | | 0101H | Erreur de recherche du fichier (erreur de décalage) |
| | | 0102H | Erreur de nombre d'octets de lecture |
| | | 0110H | Erreur de création (d'ouverture) de fichier |
| _CF_write () | LS9134 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier/de fichier spécifié dans la chaîne fixe) |
| | | 0011H | Erreur de plage de lecture de la zone LS |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0020h | Aucune carte CF |
| | | 0021h | Carte CF non valide |
| | | 0101H | Erreur de recherche du fichier (erreur de décalage) |
| | | 0104H | Erreur de création du dossier |
| | | 0108H | Erreur de mode d'écriture |
| | | 0110H | Erreur de création (d'ouverture) de fichier |
| | | 0111H | Erreur d'écriture du fichier (par exemple, espace insuffisant sur la carte CF) |

Suite

| Nom de fonction de l'éditeur | | Statut d'erreur | Cause |
|------------------------------|--------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| _CF_delete () | LS9135 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier/de fichier spécifié dans la chaîne fixe) |
| | | 0011H | Erreur de plage de lecture de la zone LS |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0020h | Aucune carte CF |
| | | 0021h | Carte CF non valide |
| | | 0112H | Erreur de suppression du fichier (Par exemple, le fichier spécifié n'existe pas. Le fichier spécifié est en lecture seule.) |
| _CF_rename () | LS9136 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier/de fichier spécifié dans la chaîne fixe) |
| | | 0011H | Erreur de plage de lecture de la zone LS |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0020h | Aucune carte CF |
| | | 0021h | Carte CF non valide |
| | | 0114H | Erreur de renommage du fichier (Par exemple, le fichier spécifié n'existe pas. Le nom de fichier existe déjà.) |
| _CF_read_csv () | LS9137 | 0001h | Erreur de paramètre |
| | | 0002h | Erreur de carte CF (Aucune carte CF, Erreur d'ouverture du fichier, Erreur de lecture du fichier) |
| | | 0003h | Erreur d'écriture |

Suite

| Nom de fonction de l'éditeur | | Statut d'erreur | Cause |
|------------------------------|--------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| __USB_dir () | LS9138 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier spécifié dans la chaîne fixe) |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0018H | Erreur de plage d'écriture de la zone LS |
| | | 0020h | Pas de stockage USB |
| | | 0021h | Stockage USB non valide |
| | | 0100h | Erreur d'ouverture du répertoire |
| __USB_read () | LS9139 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier/de fichier spécifié dans la chaîne fixe) |
| | | 0011H | Erreur de plage de lecture de la zone LS |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0018H | Erreur de plage d'écriture de la zone LS |
| | | 0020h | Pas de stockage USB |
| | | 0021h | Stockage USB non valide |
| | | 0101H | Erreur de recherche du fichier (erreur de décalage) |
| | | 0110H | Erreur de création (d'ouverture) de fichier |

| Nom de fonction de l'éditeur | | Statut d'erreur | Cause |
|------------------------------|--------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| __USB_write () | LS9140 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier/de fichier spécifié dans la chaîne fixe) |
| | | 0011H | Erreur de plage de lecture de la zone LS |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0020h | Pas de stockage USB |
| | | 0021h | Stockage USB non valide |
| | | 0101H | Erreur de recherche du fichier (erreur de décalage) |
| | | 0104H | Erreur de création du dossier |
| | | 0108H | Erreur de mode d'écriture |
| | | 0110H | Erreur de création (d'ouverture) de fichier |
| | | 0111H | Erreur d'écriture du fichier (par exemple, espace insuffisant sur le stockage USB) |

Suite

| Nom de fonction de l'éditeur | | Statut d'erreur | Cause |
|------------------------------|--------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| __USB_delete () | LS9141 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier/de fichier spécifié dans la chaîne fixe) |
| | | 0011H | Erreur de plage de lecture de la zone LS |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0020h | Pas de stockage USB |
| | | 0021h | Stockage USB non valide |
| | | 0112H | Erreur de suppression du fichier (Par exemple, le fichier spécifié n'existe pas. Le fichier spécifié est en lecture seule.) |
| __USB_rename () | LS9142 | 0010H | Données D-Script invalides (une erreur s'est produite lors de la récupération du nom de dossier/de fichier spécifié dans la chaîne fixe) |
| | | 0011H | Erreur de plage de lecture de la zone LS |
| | | 0012H | Erreur de nom de fichier (nom de chemin) |
| | | 0020h | Pas de stockage USB |
| | | 0021h | Stockage USB non valide |
| | | 0114H | Erreur de renommage du fichier (Par exemple, le fichier spécifié n'existe pas. Le nom de fichier existe déjà.) |
| __USB_read_csv () | LS9143 | 0001h | Erreur de paramètre |
| | | 0002h | Erreur de stockage USB (pas de stockage USB, erreur d'ouverture du fichier, erreur de lecture du fichier) |
| | | 0003h | Erreur d'écriture |

◆ **Mode de stockage des données**

Lorsque les données sont lues/écrites depuis/dans des périphériques et adresses à l'exécution de la fonction Lecture de fichier/Ecriture de fichier, l'ordre de stockage des données écrites (ou lues) peut être précisé.

La configuration du mode de stockage des données dans LS9130 peut modifier l'ordre de stockage. Le mode peut être sélectionné parmi quatre options : 0, 1, 2 ou 3.

REMARQUE

- Utilisez ce qui suit pour référencer LS9130
 - _CF_write() Opération de fichier CF : Ecrire le fichier
 - _CF_read() Opération de fichier CF : Lire le fichier
 - _CF_read_csv() Opération de fichier CF : Lire le fichier CSV
 - _USB_write() Opération de fichier USB : Ecrire le fichier
 - _USB_read() Opération de fichier USB : Lire le fichier
 - _USB_read_csv() Opération de fichier USB : Lire le fichier CSV
 - IO_WRITE([p:PRN],...)Opération d'impression : Envoyer
- Lors de l'écriture ou de la lecture des adresses de périphérique, au lieu d'utiliser le mode de stockage LS9130, vous pouvez utiliser les fonctions suivantes afin d'interagir avec la propriété [Mode de données texte] dans la page [Périphérique/Automate] de la fenêtre [Paramètres système].
 - _CF_dir() Opération de fichier CF : Produire la liste de fichiers
 - _USB_dir() Opération de fichier USB : Produire la liste de fichiers

• **Mode 0**

Lorsque la fonction de lecture du fichier est utilisée afin d'écrire la chaîne «ABCDEFG» dans une adresse de périphérique

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

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

- Lorsque la longueur de l'adresse de périphérique est de 16 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Lorsque la longueur de l'adresse de périphérique est de 32 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

• **Mode 1**

Par exemple, lorsque vous utilisez la fonction de lecture du fichier afin d'écrire la chaîne «ABCDEFG» dans une adresse de périphérique

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

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

- Lorsque la longueur de l'adresse de périphérique est de 16 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Lorsque la longueur de l'adresse de périphérique est de 32 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Mode 2

Par exemple, lorsque vous utilisez la fonction de lecture du fichier afin d'écrire la chaîne «ABCDEFG» dans une adresse de périphérique

[w:[#INTERNAL]LS9130] = 2

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

- Lorsque la longueur de l'adresse de périphérique est de 16 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Lorsque la longueur de l'adresse de périphérique est de 32 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Mode 3

Par exemple, lorsque vous utilisez la fonction de lecture du fichier afin d'écrire la chaîne «ABCDEFG» dans une adresse de périphérique

[w:[#INTERNAL]LS9130] = 3

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

- Lorsque la longueur de l'adresse de périphérique est de 16 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Lorsque la longueur de l'adresse de périphérique est de 32 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

IMPORTANT

- Le mode de stockage des données ne correspond pas au mode de données de chaîne dans les paramètres système. Le rapport avec le mode de données en chaîne est présenté dans le tableau suivant.

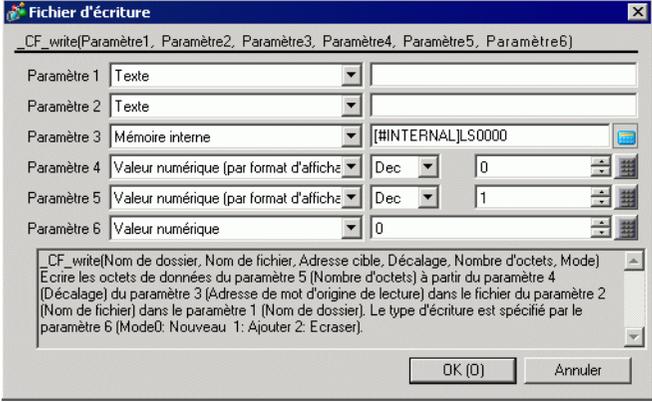
| Ordre de stockage du périphérique de données | Stockage BH/HB de l'octet de mot | Stockage BH/HB du double mot | Mode de stockage des données D-Script | Mode de données texte |
|----------------------------------------------|----------------------------------|------------------------------|---------------------------------------|-----------------------|
| Stocker selon les données de départ | Ordre HB | Ordre HB | 0 | 1 |
| | Ordre BH | | 1 | 2 |
| | Ordre HB | Ordre BH | 2 | 5 |
| | Ordre BH | | 3 | 4 |
| Stocker selon les dernières données | Ordre HB | Ordre HB | – | 3 |
| | Ordre BH | | – | 7 |
| | Ordre HB | Ordre BH | – | 8 |
| | Ordre BH | | – | 6 |

- La carte CF est limitée en écriture, veillez donc à sauvegarder régulièrement sur un autre support de stockage. En supposant que 500 Ko de données de format DOS sont écrasés, la limite est de 100000 fois.
- Si une erreur se produit lors du traitement de la carte CF ou du stockage USB, l'erreur est écrite dans le statut d'erreur de la carte CF ou du stockage USB [s:CF_ERR_STAT]/[s:USB_ERR_STAT]. Pour en savoir plus, reportez-vous à la section « Statut d'erreur de la carte CF ou du stockage USB » (page 21-112).
- Vous ne pouvez pas utiliser les signes et les caractères suivants dans des noms de dossier ou de fichier. L'utilisation de ces signes et caractères dans le nom d'un dossier ou le nom d'un fichier provoquera une erreur.

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

- Pour spécifier un dossier racine (répertoire), précisez "" (chaîne vide) en tant que nom du dossier.

■ **Ecrire le fichier**

| Elément | Description |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Vous pouvez sélectionner l'un des trois modes suivants : «Nouveau», «Ajouter» ou «Ecraser». Reportez-vous à la section «Mode de stockage des données» ci-dessous pour en savoir plus sur l'ordre de stockage des données. |
| Format | <p>_CF_write/_USB_write (noms de dossiers, noms de fichiers, adresses d'origine de lecture, décalage, nombre d'octets, mode)</p>  <p>Paramètre 1 Nom du dossier : Chaîne fixe (longueur maximale : 32 caractères à octet unique)</p> <p>Paramètre 2 Nom du fichier : Chaîne fixe, Variable interne (longueur maximale : 32 caractères à octet unique), Variable interne + Adresse temporaire</p> <p>Paramètre 3 Lire à partir de l'adresse : Adresse de périphérique, Adresse de périphérique + Adresse temporaire</p> <p>Paramètre 4 Décalage : Valeur numérique, Adresse de périphérique, Adresse temporaire (nombre maximum que vous pouvez spécifier : 65535 pour une longueur de 16 bits, 4 294 967 295 pour une longueur de 32 bits)</p> <p>Paramètre 5 Nombre d'octets : Valeur numérique, Adresse de périphérique, Adresse temporaire (longueur maximale : 1280)</p> <p>Paramètre 6 Mode : Valeur numérique, Adresse de périphérique, Adresse temporaire (valeurs disponibles : 0, 1, 2)</p> |

Aperçu du format de stockage

| Mode | Nom | Description |
|------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | Nouveau | Créer un nouveau fichier. Si un fichier portant le même nom existe, il est supprimé. |
| 1 | Ajouter | Ajoutez les données à un fichier spécifié. Si le fichier spécifié n'existe pas, un nouveau fichier est créé. |
| 2 | Ecraser | Ecrasez une partie du fichier. Si le décalage précisé est plus important que la taille du fichier, la zone supplémentaire est remplie de 0 et les données sont écrites après la zone. Si le décalage est précisé à la fin des données du fichier, l'opération revient au même que d'ajouter des données au fichier. Si le fichier n'existe pas, une erreur se produit. Pour plus d'informations sur cette erreur, reportez-vous à la section « Statut d'erreur de la carte CF ou du stockage USB » (page 21-112). |

Exemple d'expression :

[w:[#INTERNAL]LS0200] = 0// Décalage («0» lorsque le mode est «Nouveau»)

[w:[#INTERNAL]LS0202] = 100 // Nombre d'octets (100 octets)

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

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

[w:[#INTERNAL]LS0200],

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

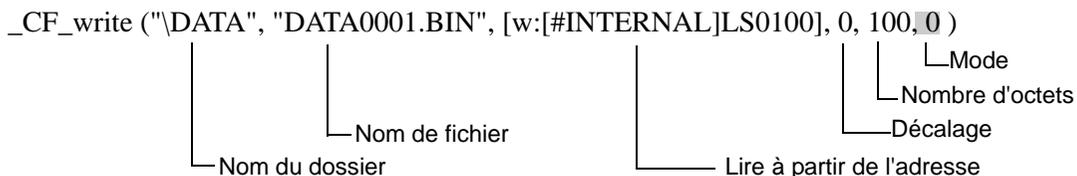
Dans l'exemple précédent, 100 octets de données sont lus depuis LS0100 et stockés dans le dossier \DATA comme DATA0001.BIN. Vous pouvez définir le nombre d'octets et le mode de façon indirecte en définissant le décalage, le nombre d'octets et le mode pour les variables internes.

IMPORTANT

- Le paramètre de décalage n'est applicable qu'en mode «Ecrasement». Le paramètre de décalage est désactivé en modes «Nouveau» et «Ajouter». Configurez la valeur de décalage à «0» en modes autres que le mode «Ecraser».
 - Lorsque le mode «Nouveau» est spécifié et qu'un fichier portant le même nom existe déjà, celui-ci est écrasé.
 - Lorsque la zone LS est spécifiée pour le «nom du fichier», l'«adresse Ecrire à partir» n'est pas comptée comme adresse D-Script.
 - Lorsqu'un périphérique automate est défini en tant qu'«Adresse d'origine de lecture», les données ne sont lues dans l'automate que lorsque la fonction est exécutée. Si une erreur se produit lors de la lecture des données, celle-ci produit une erreur de lecture de carte CF ou de stockage USB : [s:CF_ERR_STAT] ou [s:<USB_ERR_STAT]. L'erreur est effacée lorsque la lecture des données est terminée correctement.
 - Les données sont divisées en groupes et sont lues à partir de la source, même si cela dépend du nombre d'octets à lire. Donc, même s'il se produit une erreur de communication, il est possible que les données aient été partiellement écrites dans le fichier précisé.
 - Pour préciser un chemin complet pour un nom de fichier, précisez «*» (astérisque) comme nom de dossier.
Par exemple, `_CF_read ("*", "DATA\DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 10)`
-

Exemple d'expression de format de stockage expression

◆ Lorsque le mode «Nouveau» est spécifié

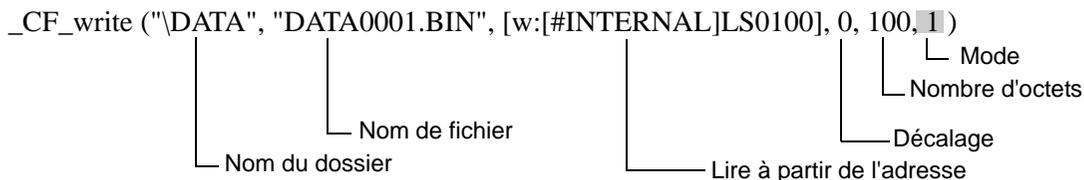


Dans l'exemple ci-dessus, 100 octets de données sont lus depuis LS0100, et le fichier DATA0001.BIN est créé dans le dossier \\DATA.

IMPORTANT

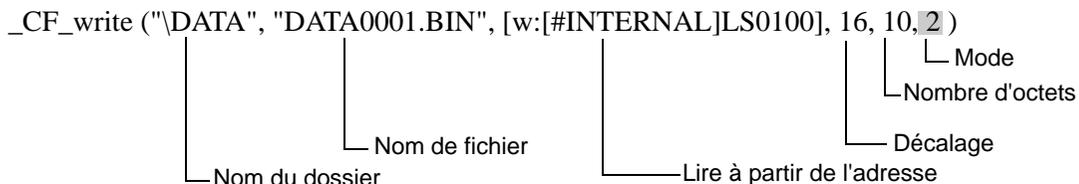
- Tandis que pour le nom de fichier, seul le «format 8.3» (un maximum de 12 caractères, 8 caractères pour le nom du fichier, le point et 3 caractères pour l'extension) peut être utilisé. Vous ne pouvez pas utiliser un nom de fichier plus long que ce format.

◆ Lorsque le mode «Ajouter» est spécifié



Si le dossier précisé (DATA0001.BIN dans l'exemple) existe déjà et que l'instruction précédente est exécutée, 100 octets de données sont lus depuis LS0100 et les zones suivantes, puis elles sont ajoutées au fichier DATA0001.BIN dans le dossier \\DATA.

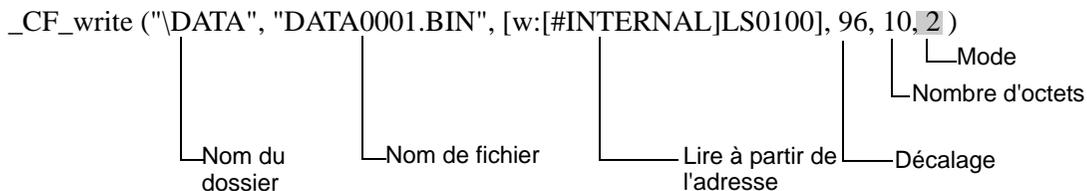
◆ Lorsque le mode «Ecraser» est spécifié (1)



Si le dossier précisé (DATA0001.BIN dans l'exemple) existe déjà et que l'instruction ci-dessus est exécutée, 10 octets de données stockées dans LS0100 et dans les zones suivantes sont lus et écrasent les 10 octets de données stockées dans le 17e octet et suivant après le décalage dans le fichier DATA0001.BIN dans le dossier \\DATA.

◆ Lorsque le mode «Ecraser» est spécifié (2)

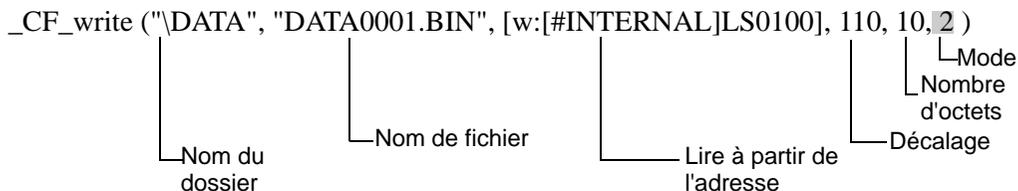
(Le fichier à écraser est inférieur à la somme de la valeur de décalage et le nombre d'octets ajoutés.)



Le fichier spécifié (DATA0001.BIN dans l'exemple) existe déjà et la taille du fichier est de 100 octets. Lorsque le décalage est configuré à 96 octets et que le nombre d'octets est configuré à 10 octets pour l'opération d'écrasement, 10 octets de données stockés dans LS0100 et dans les zones suivantes sont lus. Ensuite, les 4 premiers octets de données de lecture écrasent les 4 premiers octets de données stockés dans 97e et les octets suivants du fichier, et les 6 octets qui restent sont ajoutés à la fin des données du fichier. Le fichier résultant contient 106 octets de données.

◆ **Lorsque le mode «Ecraser» est spécifié (3)**

(Le fichier à écraser est plus petit que la valeur du décalage.)



Le fichier spécifié (DATA0001.BIN dans l'exemple) existe déjà et la taille du fichier est de 100 octets. Si le décalage est configuré à 110 octets et que le nombre d'octets est configuré à 10 octets pour l'opération d'écrasement, la zone entre le 101e octet et le 110e octet est remplie avec 0s et les 10 octets de données lues de LS0100 et les zones suivantes sont écrites dans le 111e octet et suivant. Le fichier résultant contient 120 octets de données.

IMPORTANT

- Le nombre maximum admissible de caractères pour le premier paramètre (Nom du dossier) et le deuxième paramètre (Nom de fichier) est de 32 caractères à octet unique.
- Une variable interne peut être précisée pour le deuxième paramètre (Nom de fichier). Préciser la variable interne permet l'adressage indirect d'un nom de fichier. De plus, vous pouvez utiliser jusqu'à 32 caractères à octet unique pour spécifier un nom de fichier.
Par exemple, `_CF_write ("\"DATA", [w:[#INTERNAL]LS0100], [w:[#INTERNAL]LS0200], 0, 100, 0)`
Le stockage d'un nom de fichier dans LS0100 permet l'adressage indirect d'un nom de fichier. Dans l'exemple suivant, un nom de fichier est stocké dans LS0100 et LS0106 de la façon suivante.

16 bits

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

La fin du nom de fichier doit être un caractère NUL. Le périphérique d'affichage reconnaît les données avant le caractère NUL en tant que nom de fichier.

Dans l'exemple ci-dessus, 100 octets de données sont lus depuis LS0200 et un nouveau fichier, «\DATA\DATA0001.BIN», est créé pour stocker les données.

- Tandis que pour le nom de fichier, seul le «format 8.3» (un maximum de 12 caractères, 8 caractères pour le nom du fichier, le point et 3 caractères pour l'extension) peut être utilisé. Vous ne pouvez pas utiliser les noms de fichier longs.

■ **Modifier le nom de fichier**

| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Modifie le nom du fichier. Le paramètre 1 désigne le dossier de données de la carte CF. Le paramètre 2 désigne le nom de fichier original. Le paramètre 3 désigne le nouveau nom. |
| Format | <p>_CF_rename/_USB_rename (noms de dossiers, noms de fichiers, noms de fichiers modifiés)</p> <p>Le nom de fichier peut être également désigné de façon indirecte avec l'adresse LS.</p> <div data-bbox="491 510 1146 823" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 Nom du dossier : texte fixe</p> <p>Paramètre 2 Nom du fichier : texte fixe, variable interne, variable interne + adresse temporaire</p> <p>Paramètre 3 Nom du fichier : texte fixe, variable interne, variable interne + adresse temporaire</p> |

Exemple d'expression :

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

Dans l'exemple ci-dessus, le nom de fichier «\\DATA\\DATA0001.BIN» change en «\\DATA\\DATA1234.BIN».

IMPORTANT

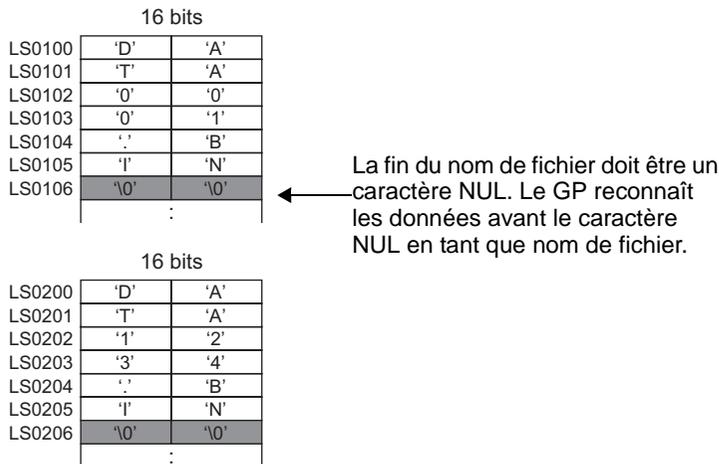
- Tandis que pour le nom de fichier, seul le «format 8.3» (un maximum de 12 caractères, 8 caractères pour le nom du fichier, le point et 3 caractères pour l'extension) peut être utilisé. Vous ne pouvez pas utiliser les noms de fichier longs.
- Le nombre maximum admissible de caractères pour le premier paramètre (Nom du fichier) et le deuxième paramètre (Nom du fichier) est de 32 caractères à octet unique.
- Une variable interne peut être précisée pour le deuxième et le troisième paramètre (noms de fichiers). Préciser la variable interne permet l'adressage indirect d'un nom de fichier. De plus, vous pouvez utiliser jusqu'à 32 caractères à octet unique pour spécifier un nom de fichier.

Exemple

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

Le stockage du nom de fichier dans LS0100 et dans LS0200 active l'adressage indirect du nom de fichier.

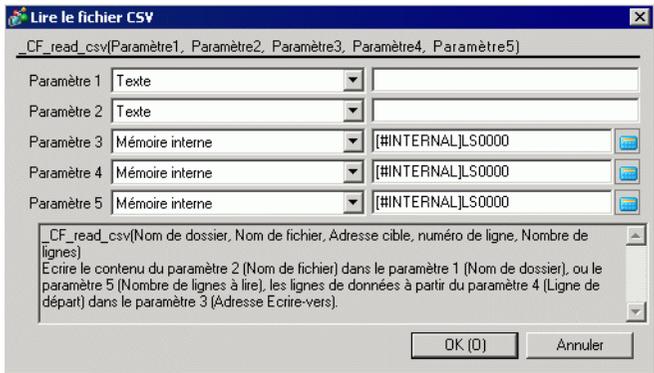
- Stockez les noms de fichier dans LS0100 et LS0106 de la façon suivante :



Dans l'instruction ci-dessus, le fichier «\\DATA\\DATA0001.BIN» est renommé en «\\DATA\\DATA1234.BIN».

- Lorsque la zone LS est spécifiée pour le «nom du fichier», il n'est pas compté comme adresse D-Script.
- Pour spécifier un dossier racine (répertoire), précisez "" (chaîne vide) en tant que nom du dossier.
- Pour préciser un chemin complet pour un nom de fichier, précisez «*» (astérisque) comme nom de dossier.

■ Lire le fichier CSV

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Lit les données dans les unités de cellule d'un fichier CSV (construit à partir d'une image de cellule délimitée par «,») et les écrit dans une adresse de mot. |
| Format | <p>_CF_read_csv/_USB_read_csv (noms de dossiers, noms de fichiers, adresses d'enregistrement, ligne de départ, nombre de lignes lues)</p>  <p>Paramètre 1 : Texte (jusqu'à 32 caractères à octet unique) Paramètre 2 : Texte (jusqu'à 32 caractères à octet unique), Variable interne, Variable interne + Adresse temporaire Paramètre 3 : Variable interne, Variable interne désignée avec le décalage Paramètre 4 : Valeur numérique (de 1 à 65 535), Variable interne, Variable temporaire Paramètre 5 : Valeur numérique (de 1 à 65 535), Variable interne, Variable temporaire</p> |

Exemple d'expression :

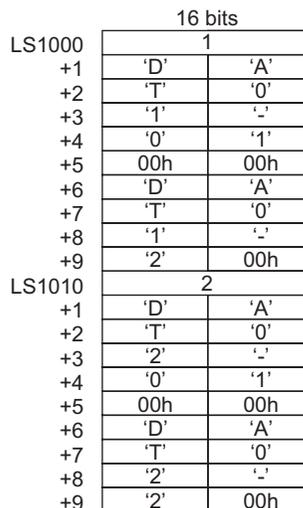
```
_CF_read_csv ("\\CSV", "SAMPLE.CSV", [w:[#INTERNAL]LS1000], 1, 2)
```

(Lors de la lecture de deux lignes de données, à compter de la première ligne du fichier [\\CSV\\SAMPLE.CSV] dans la carte CF à l'aide de la fonction «_CF_read_csv ()».)

SAMPLE.CSV

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

Lit deux lignes de données, à compter de la première ligne du fichier CSV. Lorsque le premier caractère est une valeur numérique (de «0» à «9», «-»), les données sont stockées en tant que valeur numérique. Lorsque le premier caractère est ["], les données sont traitées en tant que caractère, et «00h» est stocké à la fin de la chaîne de texte. Par exemple, lorsque vous stockez «DAT01-01», la taille des données est de 8 caractères, ce qui est un nombre pair, et 5 mots sont utilisés : Quatre mots sont utilisés pour stocker la chaîne de texte, et un mot est utilisé pour stocker «00h» à la fin. Par exemple, lorsque vous stockez «DAT01-2», la taille des données est de 7 caractères, ce qui est un nombre impair, et 4 mots sont utilisés pour stocker le texte avec «00h» stocké à la fin.

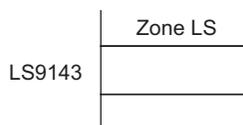
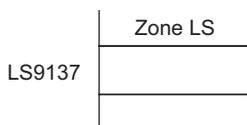


Lorsque le mode de stockage des données est 0

REMARQUE

- Lorsque le premier caractère de la cellule est une valeur numérique («0» à «9», «-»), convertit la valeur en données numériques, puis écrit les données dans le périphérique LS. La plage valide est de -32768 à 32767.
- Lorsque le premier caractère de la cellule est ["], écrit la plage avec ["] dans le périphérique LS comme données de chaîne de texte. Lorsque la taille des données de chaîne de texte est un nombre impair d'octets, «0x00» est ajouté à la fin. Lorsque la taille des données de texte est un nombre pair d'octets, «0x0000» est écrit dans l'adresse, après la dernière adresse. Vous pouvez saisir jusqu'à 32 caractères à octet unique dans une cellule.
- Lorsqu'un fichier CSV contient deux lignes de données ou plus, le nombre désiré de lignes peut être lu à compter de la ligne spécifiée. Vous pouvez saisir jusqu'à 200 caractères à octet unique dans une ligne, et jusqu'à 65535 lignes dans un fichier CSV.
- Lorsqu'une erreur se produit, le statut de l'erreur est écrit dans LS9137 (LS9143 pour le stockage USB).
- Lorsque vous écrivez des données texte de fichier CSV dans le périphérique LS, l'ordre de stockage des données dépend du mode de stockage des données.

Statut d'erreur



| Nom de fonction de l'éditeur | Zone LS | Statut d'erreur | Cause |
|----------------------------------------|-------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| _CF_read_csv ()/ _USB_read_csv () | LS9137/ LS9143 | 0000h | Terminé avec succès |
| | | 0001h | Erreur de paramètre |
| | | 0002h | Erreur de carte CF ou de stockage USB Pas de carte CF/stockage USB/ Erreur d'ouverture du fichier/ Erreur de lecture du fichier |
| | | 0003h | Erreur d'écriture/de lecture |

IMPORTANT

- Lorsque «*» est précisé pour le nom du dossier, le chemin complet peut être désigné pour le nom du fichier.
- Tandis que pour le nom de fichier, seul le «format 8.3» (un maximum de 12 caractères, 8 caractères pour le nom du fichier, le point et 3 caractères pour l'extension) peut être utilisé. Vous ne pouvez pas utiliser un nom de fichier plus long que ce format.
- La zone de périphérique LS efficace pour stocker des données importées à partir d'un fichier CSV est limitée à la zone d'utilisateur désignée (de LS20 à LS2031 et de LS2096 à LS8191).
- La durée de traitement nécessaire pour importer des données est proportionnelle au volume de données dans le fichier CSV à lire. Les objets ne sont pas actualisés tant que le traitement n'est pas terminé. (Il faut environ 10 secondes pour lire les données à partir de la 10e ligne d'un fichier CSV contenant 100 lignes, avec 40 caractères par ligne.)
- Contrairement à la fonction «_CF_read (/)_USB_read()», le statut n'est pas enregistré dans [s:CF_ERR_STAT]/[s:USB_ERR_STAT] immédiatement après l'exécution de la fonction. (Dans certains cas, des valeurs non définies peuvent être stockées.)
- Assurez-vous d'insérer "[" au début et à la fin des chaînes de texte qui commencent avec un nombre.

Par exemple :

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

■ Lire le fichier

| Elément | Description |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Lit le nombre précisé d'octets de données dans le fichier après le décalage précisé et l'écrit dans l'adresse de destination. Reportez-vous à la section «Mode de stockage des données» ci-dessous pour en savoir plus sur l'ordre de stockage des données. |
| Format | <p><code>_CF_read/_USB_read</code> (noms de dossiers, noms de fichiers, adresses Lire-à-partir, décalage, nombre d'octets)</p> <div data-bbox="491 479 1146 850" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 Nom du dossier : Chaîne fixe (longueur maximale : 32 caractères à octet unique)</p> <p>Paramètre 2 Nom du fichier : Chaîne fixe, Variable interne, Variable interne + Adresse temporaire (longueur maximale : 32 caractères à octet unique)</p> <p>Paramètre 3 Adresse Ecrire vers : Adresse de périphérique, Adresse de périphérique + Adresse temporaire</p> <p>Paramètre 4 Décalage : Valeur numérique, Adresse de périphérique, Adresse temporaire (nombre maximum que vous pouvez spécifier : 65535 pour une longueur de 16 bits, 4 294 967 295 pour une longueur de 32 bits)</p> <p>Paramètre 5 Nombre d'octets : Valeur numérique, Adresse de périphérique, Adresse temporaire (longueur maximale : 1280)</p> |

Exemple d'expression :

Pour lire 16 octets de données dans le fichier spécifié lorsque le décalage est de 16 :

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

Dans l'exemple ci-dessus, les 16 octets de données à compter du 17e octet dans le fichier «\\DATA\\DATA0001.BIN» sont écrits dans la zone à compter de LS0100.

IMPORTANT

- Tandis que pour le nom de fichier, seul le «format 8.3» (un maximum de 12 caractères, 8 caractères pour le nom du fichier, le point et 3 caractères pour l'extension) peut être utilisé. Vous ne pouvez pas utiliser les noms de fichier longs.
- Le nombre maximum admissible de caractères pour le premier paramètre (Nom du fichier) et le deuxième paramètre (Nom du fichier) est de 32 caractères à octet unique.
- Une variable interne peut être précisée pour le deuxième paramètre (Nom de fichier). Préciser la variable interne permet l'adressage indirect d'un nom de fichier. De plus, vous pouvez utiliser jusqu'à 32 caractères à octet unique pour spécifier un nom de fichier.

Exemple

Pour lire 10 octets de données stockées dans un fichier lorsque le fichier est précisé dans LS0100 et suivant et que le décalage est de 0 :

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

Le stockage d'un nom de fichier dans LS0100 permet l'adressage indirect d'un nom de fichier. Dans l'exemple suivant, un nom de fichier est stocké dans LS0100 et LS0106 de la façon suivante.

16 bits

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

La fin du nom de fichier doit être un caractère NUL. Le périphérique d'affichage reconnaît les données avant le caractère NUL en tant que nom de fichier.

Dans l'exemple ci-dessus, les 10 octets de données au début du fichier «DATA\DATA0001.BIN» sont lus et écrits dans la zone à compter de LS0200.

- Le nombre d'octets lus avec succès est écrit dans les octets de lecture de carte CF/stockage USB [s:CF_READ_NUM]/[s:USB_READ_NUM]. Pour en savoir plus, reportez-vous à la section «21.11.5 Opération de fichier CF/ Opération de fichier USB Statut d'erreur de la carte CF ou du stockage USB» (page 21-112).
- Les variables internes désignées dans le «nom de fichier» et l'«adresse Ecrire vers» ne sont pas comptées en tant qu'adresses D-Script.
- Lorsqu'un automate est précisé pour l'Adresse Ecrire vers, il faut plus de temps pour écrire les données dans l'automate au fur et à mesure que le nombre de mots (octets) augmente. Il peut falloir plusieurs secondes, selon le nombre de mots.
- Si les données lues à partir du fichier dépassent la plage de périphérique désignée de l'automate, une erreur de communication se produit. Dans ce cas, vous devez mettre l'alimentation de l'automate hors tension, puis sous tension pour réarmer l'automate après l'erreur.

IMPORTANT

- Lorsqu'un automate est précisé comme destination, les valeurs ne sont pas écrites immédiatement en raison de la durée de transmission GP à automate.

Exemple

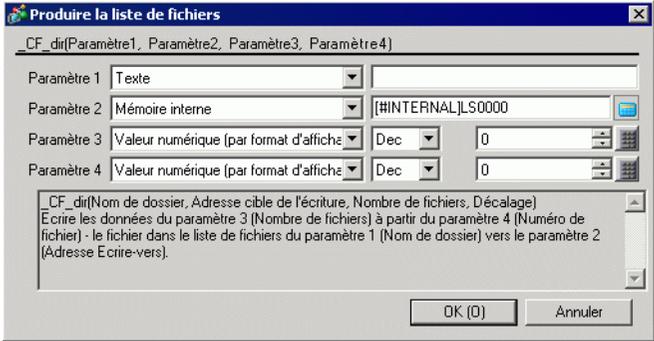
Dans le script suivant, l'instruction (1) lit 10 octets de données à partir du fichier et écrit les données dans [w:D0100]. Toutefois, les données n'ont toujours pas été écrites dans [w:[PLC1]D0100] lors de l'exécution de l'instruction (2) en raison de la durée de la transmission.

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

Dans un tel cas, stockez les données une fois qu'elles sont dans la zone LS, puis exécutez la deuxième instruction, comme suit :

```
_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
```

■ Produire la liste de fichiers

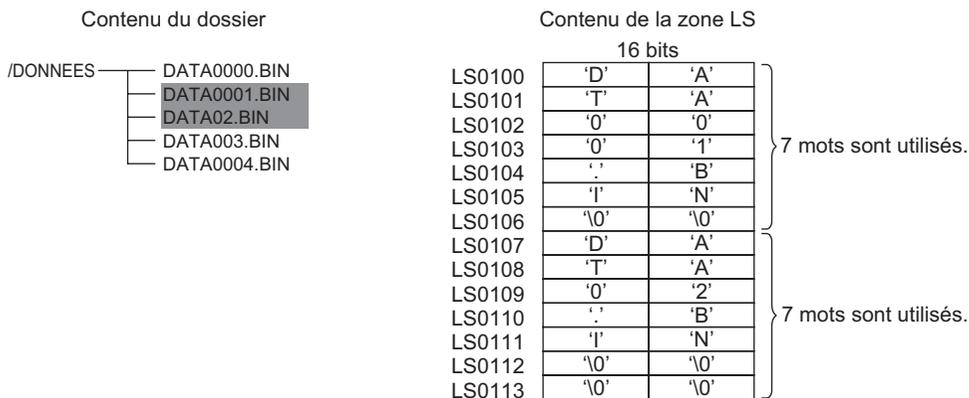
| Elément | Description |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>La liste de fichiers qui existent dans le dossier spécifié est écrite dans la mémoire interne. Le paramètre 1 indique le dossier de données de carte CF. Le paramètre 4 indique le décalage utilisé pour sélectionner un(des) fichier/fichiers qui se trouve(nt) dans ce dossier. Le paramètre 3 indique le nombre de fichiers sélectionnés dans ce dossier. Le paramètre 2 spécifie la zone LS dans laquelle les fichiers seront écrits. Lorsque le décalage est précisé comme «0», la liste commence au premier fichier (de départ).</p> |
| Format | <p>_CF_dir/_USB_dir (noms de dossiers, adresses d'enregistrement, nombre de fichiers, décalage)</p>  <p>Paramètre 1 Nom du dossier : Texte fixe (longueur maximale : 32 caractères à octet unique)</p> <p>Paramètre 2 Adresse Ecrire vers : Variable interne, Variable interne désignée avec le décalage</p> <p>Paramètre 3 Nombre de fichiers : Valeur numérique, Adresse de périphérique, Adresse temporaire (longueur maximale : 32)</p> <p>Paramètre 4 Valeur numérique, Adresse de périphérique, Adresse temporaire</p> |

Exemple d'expression :

Pour produire une liste de fichiers qui comprend deux fichiers lorsque le décalage est de 1 (deuxième fichier) :

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

Lorsqu'on exécute l'instruction ci-dessus et les fichiers suivants existent dans le dossier DATA, les noms de fichier «DATA0001.BIN» et «DATA02.BIN» sont écrits dans LS0100 et antérieurs.



IMPORTANT

- Lorsque le décalage est précisé comme «0», la liste commence au premier fichier (de départ).
- Tandis que pour le nom de fichier, seul le «format 8.3» (un maximum de 12 caractères, 8 caractères pour le nom du fichier, le point et 3 caractères pour l'extension) peut être utilisé. Vous ne pouvez pas utiliser les noms de fichier longs.
- Si le dossier précisé ne possède pas assez de fichiers, tel que cela est précisé, la zone LS restante sera remplie de caractères Null («\0»).
- Si un nom de fichier contient moins de 12 caractères, les positions vides sont remplies avec des caractères NUL («\0»).
- Lorsque vous précisez un nom de dossier, comme «\DATA*.*», assurez-vous d'ajouter «*.*». Les astérisques *.* indique qu'il faut répertorier tous les fichiers.
- Le nombre de fichiers effectivement répertoriés sont écrits dans les fichiers répertoriés de la carte CF ou du stockage USB [s:CF_FILELIST_NUM]/ [s:USB_FILELIST_NUM].
Pour en savoir plus, reportez-vous à la section « Statut d'erreur de la carte CF ou du stockage USB» (page 21-112)
- Les adresses Ecrire vers LS sont comptées comme adresses D-Script.
- Les noms de fichier ne sont pas triés quand ils sont écrits dans la zone LS. Ils sont écrits en ordre de création (l'ordre de l'entrée FAT).
- Vous pouvez créer la liste en spécifiant une extension de fichier. Pour répertorier les fichiers avec une extension déterminée, utilisez un format tel que «\DATA*.BIN». Toutefois, vous ne pouvez pas utiliser «*» dans un nom de fichier.

■ **Supprimer le fichier**

| Elément | Description |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Supprime le fichier spécifié de la carte CF. Le paramètre 1 indique le dossier de données de carte CF. Le paramètre 2 indique le nom de fichier à supprimer. |
| Format | <p>_CF_delete/_USB_delete (noms de dossiers, noms de fichiers) Le nom de fichier peut être également désigné de façon indirecte avec l'adresse LS.</p> <div data-bbox="492 473 1149 784" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Paramètre 1 Nom du dossier : texte fixe</p> <p>Paramètre 2 Nom du fichier : texte fixe, variable interne, variable interne + adresse temporaire</p> |

Exemple d'expression :

_CF_delete ("\\DATA", "DATA0001.BIN")

L'exemple ci-dessus supprime le fichier «\\DATA\\DATA0001.BIN».

IMPORTANT

- Tandis que pour le nom de fichier, seul le «format 8.3» (un maximum de 12 caractères, 8 caractères pour le nom du fichier, le point et 3 caractères pour l'extension) peut être utilisé. Vous ne pouvez pas utiliser les noms de fichier longs.
- Le nombre maximum admissible de caractères pour le premier paramètre (Nom du fichier) et le deuxième paramètre (Nom du fichier) est de 32 caractères à octet unique.
- Une variable interne peut être précisée pour le deuxième paramètre (Nom de fichier). Préciser la variable interne permet l'adressage indirect d'un nom de fichier. De plus, vous pouvez utiliser jusqu'à 32 caractères à octet unique pour spécifier un nom de fichier.

Dans l'exemple suivant, un nom de fichier est stocké dans LS0100 et LS0106 de la façon suivante.

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

La fin du nom de fichier doit être un caractère NUL. Le périphérique d'affichage reconnaît les données avant le caractère NUL en tant que nom de fichier.

Dans l'exemple ci-dessus, le fichier «\DATA\DATA0001.BIN» est supprimé.

- Pour spécifier un dossier racine (répertoire), précisez "" (chaîne vide) en tant que nom du dossier.
- Lorsque la zone LS est spécifiée pour le «nom de fichier», les «adresses Ecrire vers» ne sont pas comptées en tant qu'adresses D-Script.
- Pour préciser un chemin complet pour un nom de fichier, précisez «*» (astérisque) comme nom de dossier.

21.11.6 Opération d'imprimante

| Opération d'imprimante | Résumé de la fonction |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Paramètres d'étiquette « ■ Paramètres d'étiquette » (page 21-139) Désigné à partir des variables de contrôle et de statut.</p> |
| | <p>Envoyer « ■ Envoyer » (page 21-141) Produit le nombre désigné d'octets vers le port COM.</p> |

IMPORTANT

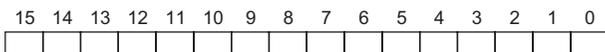
- COM1 ou USB/PIO (USB-PIO) sont des ports qui peuvent être utilisés comme fonction Opération d'imprimante.

■ Paramètres d'étiquette

Contrôle

Le contrôle (PRN_CTRL) est une variable utilisée pour effacer le tampon d'envoi et le statut d'erreur. Cette variable est en écriture seule.

- Résumé du contrôle (PRN_CTRL)



| Bit | Contenu |
|-----|------------------------------|
| 15 | Réservé |
| 14 | |
| 13 | |
| 12 | |
| 11 | |
| 10 | |
| 9 | |
| 8 | |
| 7 | |
| 6 | |
| 5 | |
| 4 | |
| 3 | |
| 2 | 1: Erreur d'effacement |
| 1 | Réservé |
| 0 | 1: Effacer le tampon d'envoi |

IMPORTANT

- Lorsqu'un mot est sélectionné et que deux bits ou plus sont configurés simultanément, le traitement est exécuté dans l'ordre suivant :

Effacer l'erreur



Effacer le tampon d'envoi

- N'utilisez pas les bits réservés. Ne configurez que les bits nécessaires.

Statut

La variable de statut (PRN_STAT) est utilisée afin de vérifier la présence/l'absence des données dans le tampon d'envoi et pour obtenir le statut d'erreur. Cette variable de statut est en écriture seule.

- Contenu de la variable de statut (PRN_STAT)

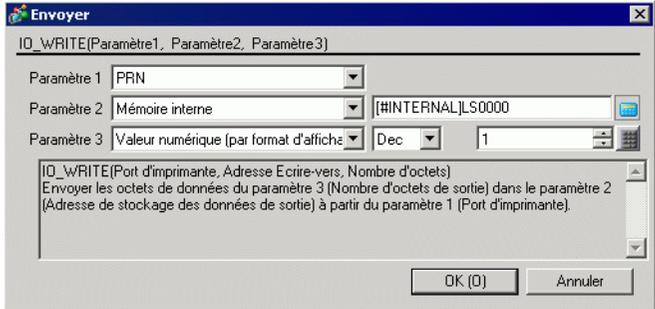
| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | | | | | | | |

| Bit | Contenu |
|-----|------------------------------------------------------------------------------------------------------|
| 15 | Réservé |
| 14 | Etat du signal ERROR I/F de l'imprimante Erreur d'imprimante (Saisie) : 0: Erreur 1: Normal |
| 13 | Etat du signal SLCT I/F de l'imprimante Sélectionner (Saisie) : 0: Hors ligne 1: En ligne |
| 12 | Etat du signal PE I/F de l'imprimante Carte (Saisie) : 0: Normal 1: Manque de papier |
| 11 | Riservato |
| 10 | |
| 9 | |
| 8 | |
| 7 | |
| 6 | |
| 5 | |
| 4 | |
| 3 | |
| 2 | |
| 1 | 0: Normal 1: Erreur d'envoi |
| 0 | 0: Des données existent dans le tampon d'envoi 1: Le tampon d'envoi est vide |

IMPORTANT

- Si le tampon d'envoi est trop plein, une erreur se produit. Lorsque cette erreur se produit, le bit d'erreur de transmission est activé.
- Le tampon d'envoi est de 8 192 octets.
- Les bits réservés peuvent être affectés dans le futur. Donc, assurez-vous de ne vérifier que les bits nécessaires.

■ Envoyer

| Elément | Description |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Produit le nombre désigné d'octets vers le port COM. Les données sont produites, peu importe le type d'imprimante précisé. |
| Format | <p>IO_WRITE ([p:PRN], Adresse de stockage des données de sortie, Nombre d'octets de sortie)</p>  <p>Paramètre 1 : [p:PRN] Paramètre 2 : Variable interne Paramètre 3 : Valeur entière, Adresse de périphérique, Adresse temporaire</p> |

IMPORTANT

- La valeur maximale que vous pouvez assigner au paramètre 3 est de 1024. Même si vous précisez des valeurs supérieures à 1024, seuls 1024 octets de données sont produits depuis le port COM.

Exemple d'expression 1 :

```
IO_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 10)
```

Dans l'exemple ci-dessus, 10 octets de données stockées dans les zones LS1000 et suivantes sont produits depuis le port COM.

Exemple d'expression 2 :

```
IO_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS0800])
```

Dans l'exemple ci-dessus, les données stockées dans les zones LS1000 et suivantes sont produits depuis le port COM. Le nombre d'octets est le même que ce qui a été écrit dans LS0800.

Exemple d'expression 3 :

```
IO_WRITE ([p:PRN], [w:[#INTERNAL]LS 1000], [t:0010])
```

Dans l'exemple ci-dessus, les données stockées dans les zones LS1000 et suivantes sont produits depuis le port COM. Le nombre d'octets qui correspondent à ceux écrits dans l'adresse temporaire [t:0010].

Mode de stockage des données

Lorsque des données sont lues à partir de périphériques et d'adresses à l'exécution de la fonction Opération de port COM, vous pouvez préciser l'ordre de stockage des données de lecture. La configuration du mode de stockage des données dans LS9130 peut modifier l'ordre de stockage. Le mode peut être sélectionné parmi quatre options : 0, 1, 2 ou 3.

◆ Mode 0

Par exemple, lorsque vous utilisez la fonction d'opération du port COM pour lire la chaîne «ABCDEFG» dans une adresse de périphérique

[w:[#INTERNAL]LS9130] = 0

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

- Lorsque la longueur de l'adresse de périphérique est de 16 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Lorsque la longueur de l'adresse de périphérique est de 32 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

◆ Mode 1

Par exemple, lorsque vous utilisez la fonction d'opération du port COM pour lire la chaîne «ABCDEFG» dans une adresse de périphérique

[w:[#INTERNAL]LS9130] = 1

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

- Lorsque la longueur de l'adresse de périphérique est de 16 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Lorsque la longueur de l'adresse de périphérique est de 32 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

◆ Mode 2

Par exemple, lorsque vous utilisez la fonction d'opération du port COM pour lire la chaîne «ABCDEFG» dans une adresse de périphérique

[w:[#INTERNAL]LS9130] = 2

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

- Lorsque la longueur de l'adresse de périphérique est de 16 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Lorsque la longueur de l'adresse de périphérique est de 32 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

◆ **Mode 3**

Par exemple, lorsque vous utilisez la fonction d'opération du port COM pour lire la chaîne «ABCDEFG» dans une adresse de périphérique

[w:[#INTERNAL]LS9130] = 3

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

- Lorsque la longueur de l'adresse de périphérique est de 16 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

- Lorsque la longueur de l'adresse de périphérique est de 32 bits

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

← Ecrivez «0» lorsque les données à stocker produisent un nombre impair d'octets.

IMPORTANT

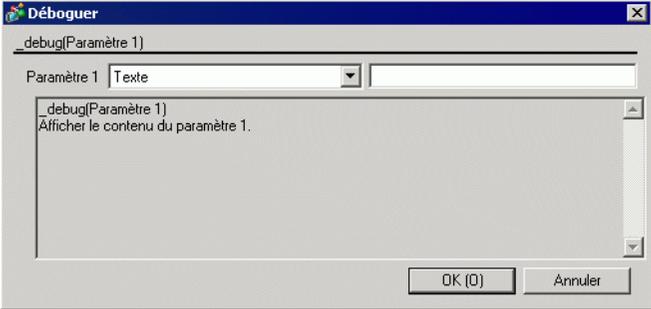
- Le mode de stockage des données ne correspond pas au mode de données de chaîne dans les paramètres système. Le rapport avec le mode de données en chaîne est présenté dans le tableau suivant.

| Ordre de stockage du périphérique de données | Stockage BH/HB de l'octet de mot | Stockage BH/HB du double mot | Mode de stockage des données D-Script | Mode de données texte |
|----------------------------------------------|----------------------------------|------------------------------|---------------------------------------|-----------------------|
| Stocker selon les données de départ | Ordre HB | Ordre HB | 0 | 1 |
| | Ordre BH | | 1 | 2 |
| | Ordre HB | Ordre BH | 2 | 5 |
| | Ordre BH | | 3 | 4 |
| Stocker selon les dernières données | Ordre HB | Ordre HB | – | 3 |
| | Ordre BH | | – | 7 |
| | Ordre HB | Ordre BH | – | 8 |
| | Ordre BH | | – | 6 |

21.11.7 Autres

| Autres | Résumé de la fonction |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Fonction de débogage « ■ Fonction de débogage » (page 21-144) Affiche l'adresse ou le texte désigné à l'écran afin de le déboguer.</p> |
| | <p>Déclenchement de l'application « ■ Déclenchement de l'application » (page 21-146) Exécute la plage spécifiée et démarre l'application.</p> |
| | <p>WinGP, quitter « ■ Quitter WinGP » (page 21-148) Quitter WinGP.</p> |

■ Fonction de débogage

| Élément | Description |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Affiche l'adresse ou le texte désigné à l'écran afin de le déboguer. Après avoir terminé le débogage et retiré la coche de la case [Activer la fonction de débogage], aucun script n'est supprimé. L'écran de débogage n'apparaît pas. |
| Format | <p><code>_debug (Paramètre 1)</code></p>  <p>Paramètre 1 : Texte (jusqu'à 32 caractères à octet unique, 16 caractères à deux octets)</p> |

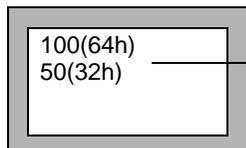
Contenu du paramètre 1

| Paramètre 1 | Format | Description |
|--------------------------------------|------------------------|--------------------------------------------------------------------------------------------------------|
| Texte | _debug («ABC») | Affiche le texte qui se trouve entre « ». Le texte peut contenir jusqu'à 32 caractères à octet unique. |
| Adresse de mot ou adresse temporaire | _debug (w:[PLC1]D1000) | Affiche la valeur de l'adresse de mot ou de l'adresse temporaire configuré. |
| Présentation de ligne | _debug (_CRLF) | Déplace le curseur au début de la ligne suivante. |
| Retour de chariot | _debug (_CR) | Déplace le curseur au début de la même ligne. |

◆ Exemple d'expression 1 :

Le script suivant affiche la valeur de l'adresse de mot.

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



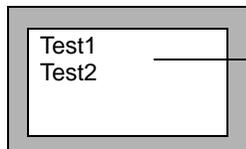
L'affichage apparaît au format suivant :
***** (**h)

Décimal Hexadécimal

◆ Exemple d'expression 2 :

Le script suivant affiche une présentation de ligne et un texte.

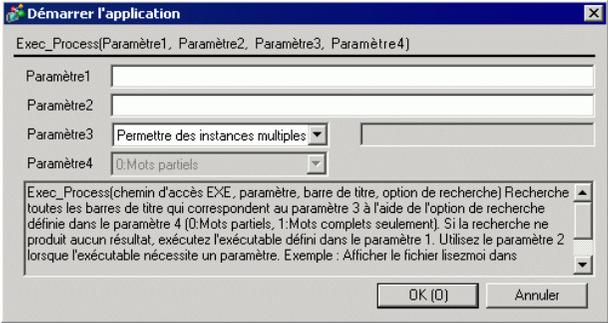
```
_debug ("Test1")
_debug (_CRLF)
_debug ("Test2")
```



Pour dérouler une ligne et afficher «Test2».

■ Déclenchement de l'application

Cette fonction ne fonctionne pas sur les modèles autres que l'IPC Series.

| Elément | Description |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | <p>Exécute la plage spécifiée et démarre l'application. Vous pouvez préciser des paramètres comme les paramètres de démarrage et la surveillance du démarrage de multiplex.</p> |
| Format | <p>Exec_Process (Paramètre 1, Paramètre 2, Paramètre 3, Paramètre 4)</p>  <p>Paramètre 1 Chemin d'accès EXE :Saisissez le chemin d'accès absolu du fichier exécutable (.exe) de l'application que vous souhaitez démarrer. Vous pouvez saisir jusqu'à 255 caractères.</p> <p>Paramètre 2 Paramètre :Saisissez l'argument de démarrage du fichier exécutable. Vous pouvez saisir jusqu'à 255 caractères.</p> <p>Paramètre 3 Titre de fenêtre : Si vous ne souhaitez pas permettre des instances multiples, sélectionnez «Ne pas permettre des instances multiples» et saisissez le [Titre de fenêtre]. Vous pouvez saisir jusqu'à 63 caractères. L'application ne peut pas être lancée si une autre fenêtre portant le même titre que dans [Titre de fenêtre] est trouvée. Des instances multiples sont permises si vous sélectionnez l'option [Permettre des instances multiples] ou si le [Titre de fenêtre] n'est pas précisé.</p> <p>Paramètre 4 Ne trouver que des titres de fenêtre entiers : Cette option n'est activée que si vous sélectionnez le paramètre3 - «Ne pas permettre des instances multiples». Lorsque vous sélectionnez l'option «0 : Mots partiels», l'application précisée n'est pas exécutée si l'on trouve une fenêtre ayant un titre qui correspond partiellement au [Titre de fenêtre]. Lorsque vous sélectionnez l'option «1 : Mots entiers uniquement», l'application précisée n'est pas exécutée si l'on trouve une fenêtre ayant un titre qui correspond exactement au [Titre de fenêtre].</p> |

REMARQUE

- Paramètre1 requiert du texte (chemin EXE). Une erreur se produit si vous ne saisissez pas du texte.
- Cette fonction ne fonctionne pas sur les modèles autres que l'IPC Series.

Méthode d'entrée du paramètre1 (chemin EXE)

Il existe 3 façons pour saisir le chemin EXE :

La description suivante donne un exemple d'exécution du fichier sample.exe dans C:\Documents and Settings\user\Local Settings\Temp

1. Spécification du chemin d'accès complet

Par exemple, C:\Documents and Settings\user\Local Settings\Temp\sample.exe

2. Nom EXE uniquement

Si le fichier exécutable se trouve dans un dossier précisé comme chemin d'accès dans les paramètres d'environnement sur une machine cible IPC Series.

Par exemple, sample.exe

(Démarrer si le chemin d'accès est le suivant : C:\Documents and Settings\user\Local Settings\Temp)

3. Définir un chemin qui contient une variable d'environnement

Si le fichier exécutable se trouve dans un dossier précisé dans les paramètres d'environnement sur une machine cible IPC Series.

Par exemple, %TEMP%\sample.exe

(Démarrer si le paramètre d'environnement est configuré à TEMP=C:\Documents and Settings\user\Local Settings\Temp)

Exemple d'expression 1 :

Permettre des instances multiples. (Démarez Bloc-notes et afficher le fichier Readme.txt)

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

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

Exemple d'expression 2 :

Ne pas permettre des instances multiples :

Mots partiels (Démarez Bloc-notes et afficher le fichier Readme.txt) Exec_Process ("C:\WINDOWS\SYSTEM32\notepad.exe", "D:\TEMP\Readme.txt", "Readme", 0)

Exemple d'expression 3 :

Ne pas permettre des instances multiples : Mots complets seulement (Démarez Bloc-notes et afficher le fichier Readme.txt)

```
Exec_Process
```

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

Exemple d'expression 4 :

Ne pas permettre des instances multiples : Mots partiels (Démarez Bloc-notes)

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

Exemple d'expression 5 :

Aucun paramètre (Démarez Bloc-notes)

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

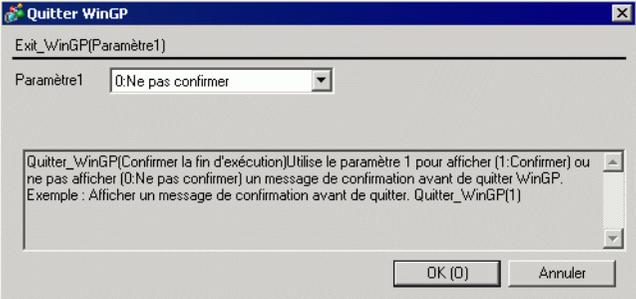
Exemple d'expression 6 :

Paramètre multiple (Démarez sample.exe)

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

■ Quitter WinGP

Cette fonction ne fonctionne pas sur les modèles autres que l'IPC Series.

| Élément | Description |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Quitter WinGP. Vous pouvez afficher un message d'acquiescement lorsque vous quittez WinGP. |
| Format | Exit_WinGP(Paramètre1) <div style="text-align: center;">  </div> <p>Paramètre 1 Nom de dossier : Sélectionnez «0: Ne pas confirmer» ou «1: Confirmer».</p> |

REMARQUE

- Paramètre1 requiert du texte (chemin EXE). Une erreur se produit si vous ne saisissez pas du texte.
- Cette fonction ne fonctionne pas lorsque vous transférez le script «Quitter WinGP» vers des modèles autres que l'IPC Series.

Exemple d'expression :

Affichage d'un message d'acquiescement lorsque vous quittez WinGP.

Exit_WinGP(1)

21.11.8 Expressions conditionnelles

| Expressions conditionnelles | Résumé de la fonction |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Expression de description if-endif if-else-endif loop-endloop break return | if - endif  « ■ if-endif » (page 21-149) Si la condition «if» entre parenthèses «()», est vraie, l'expression à la suite de l'instruction «if ()» est exécutée. |
| | if - else - endif  « ■ if - else - endif » (page 21-149) Si la condition «if» entre parenthèses «()», est vraie, l'expression à la suite de l'instruction «if ()» est exécutée. Si la condition est fausse, l'expression «else» est exécutée. |
| | loop - endloop  « ■ loop - endloop » (page 21-150) Le traitement de boucles est répété selon le nombre stocké dans les adresses temporaires désignées entre parenthèses «()» qui se trouvent après «loop». |
| | break  « ■ break » (page 21-153) Arrête l'opération de boucle lors de l'exécution de l'équation de boucle (). |
| | return  « ■ return » (page 21-153) Exécute de nouveau, depuis le début. Ne peut être utilisé que dans un script étendu. |

■ if-endif

Si la condition «if» entre parenthèses «()», est vraie, l'expression à la suite de l'instruction «if ()» est exécutée.

REMARQUE

- Vous ne pouvez pas utiliser le caractère d'affectation «=» dans une expression conditionnelle.

■ if - else - endif

Si la condition «if» entre parenthèses «()», est vraie, l'expression à la suite de l'instruction «if ()» est exécutée. Si la condition est fausse, l'expression «else» est exécutée.

REMARQUE

- Vous ne pouvez pas utiliser le caractère d'affectation «=» dans une expression conditionnelle.

■ loop - endloop

Le traitement de boucles est répété selon le nombre stocké dans les adresses temporaires désignées entre parenthèses «()» qui se trouvent après «loop».

Boucle infinie

La boucle est infinie lorsque aucune instruction n'apparaît entre parenthèses ().

Vous pouvez utiliser des boucles infinies dans les scripts étendus.

Exemple d'expression :

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

REMARQUE

- Le format de boucle () est comme suit :
 Par exemple :

```

loop (nombre de boucles)// Définit l'adresse temporaire qui stocke le
nombre de boucles.
{
    Equation d'action
    break // Utiliser pour quitter la boucle à mi-chemin (optionnel)
} endloop // Définit la fin de la boucle
    
```
- Seule une adresse de mot temporaire peut être saisie (entre parenthèses). (Par exemple, loop ([t:000]))
- Vous ne pouvez pas utiliser «loop ()» pour une équation de déclenchement.
- La valeur d'adresse de mot temporaire utilisée pour définir le nombre de boucles diminue après chaque boucle. Lorsque la valeur est modifiée à 0, l'opération de la boucle se termine. Si la valeur d'adresse de mot temporaire définie pour le nombre de boucles est modifiée, la boucle sera désormais sans fin. L'adresse de mot temporaire utilisée est désignée comme globale. Donc, utiliser cette adresse de mot temporaire simultanément à d'autres fins peut produire une boucle infinie.
- Jusqu'à ce que l'opération de boucle se termine, les affichages d'écran des objets, et ainsi de suite, ne sont pas mis à jour ou actualisés.
- loop () peut également être imbriquée. Lorsqu'elle est imbriquée, loop () la plus à l'intérieur est sautée par la commande «break».

```

loop ([t:0000]) // boucle 1
{
    loop ([t:0001]) // boucle 2
    {
        break // Echappement de la boucle 2
    }endloop
}endloop

break // Echappement de la boucle 1
}endloop
    
```
- Si l'opération de boucle se termine sans utiliser la commande d'échappement, la valeur de l'adresse de mot temporaire devient 0.

REMARQUE

- La plage disponible pour la valeur de l'adresse de mot temporaire est différente selon le format des données (Bin, BCD), la longueur de bit, et le code +/- utilisés. Si le code +/- a été configuré et que l'adresse de mot temporaire est devenue une valeur négative, la condition est jugée au début de la boucle et le traitement de la boucle s'arrête.
- N'UTILISEZ PAS un périphérique dans la formule de boucle. Utilisez plutôt l'adresse de zone utilisateur/LS interne de l'afficheur ou une adresse de mot temporaire. Par exemple, la description suivante effectue l'écriture des données vers l'automate plusieurs fois pendant une courte durée (100 fois dans l'exemple suivant). Cela peut provoquer une erreur système puisque le traitement de communication (la durée nécessaire pour écrire vers l'automate) ne peut pas être effectué à cette vitesse.

Par exemple :

```
[t:0000] = 100 // 100 boucles
loop ([t:0000])
{
  [w:[PLC1]D0200] = [w:[#INTERNAL]LS0100] // Ecrire dans D0200
  [w:[#INTERNAL]LS0100] = // Incréments LS0100
  [w:[#INTERNAL]LS0100] + 1
}endloop
```

Effectuez les modifications suivantes :

```
[t:0000] = 100 // 100 boucles
loop ([t:0000])
{
  [w:[#INTERNAL]LS0200] =
  [w:[#INTERNAL]LS0100] // Ecrire dans D0200
  [w:[#INTERNAL]LS0100] = // Incréments LS0100
  [w:[#INTERNAL]LS0100] + 1
}endloop
[w:[PLC1]D0200]=[w:[#INTERNAL]LS0200] // contenu de LS0200,
écrire dans D0200
```

- Une erreur se produit si vous utilisez «loop» ou «break» en tant que nom de fonction pour une fonction D-Script.

■ break

Quitte l'opération de boucle à mi-chemin de l'opération loop ().

REMARQUE

- La commande «break» ne peut être utilisée que dans la section { } de loop ().
 - Les scripts ne fonctionneront pas correctement si vous utilisez la commande «break» dans des expressions if { }.
-

■ return

Lorsque la «Fonction définie par l'utilisateur» inclut «return», le traitement de la fonction est terminé et le contrôle retourne à l'appelant de la fonction.

Lorsque l'exécution (la fonction principale) inclut la commande «return»

Le traitement de la fonction principale est abandonné et redémarré au début de la fonction principale.

REMARQUE

- Vous ne pouvez pas utiliser le caractère d'affectation «=» dans une expression conditionnelle.
-

Exemple d'expression :

```
[w:[#INTERNAL]LS0100]=[w:[#INTERNAL]LS0200]>> 8) & 0xFF
if ([w:[#INTERNAL]LS0100]==0)           // Lorsque LS0100 est «0», le traitement n'est
                                         plus exécuté
{
    set([b:[#INTERNAL]LS005000])        // Configure l'adresse de bit pour l'affichage
                                         d'erreur
    return                               // Fin
}
endif
```

21.11.9 Comparaison

| Comparaison | Résumé de la fonction |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Comparaison ET logique (AND) OU logique (OR) Négation (not) inférieur à (<) inférieur ou égal à (<=) différent de (<>) supérieur à (>) supérieur ou égal à (>=) Equivalent (==) | ET logique (and)  « ■ ET logique (and) » (page 21-154) N1 et N2 : Vrai si N1 et N2 sont ACTIVES. |
| | OU logique (or)  « ■ OU logique (or) » (page 21-154) N1 ou N2 : Vrai si N1 ou N2 sont ACTIVES. |
| | Négation (non)  « ■ Négation (not) » (page 21-154) not N1 : Devient 0 si N1 est 1, et 1 si N1 est 0. |
| | Inférieur à (<)  « ■ inférieur à (<) » (page 21-155) Vrai si N1 est supérieur à N2 ($N1 < N2$). |
| | Inférieur ou égal à (=)  « ■ Inférieur ou égal à (<=) » (page 21-155) Vrai si N1 est inférieur ou égal à N2 ($N1 <= N2$). |
| | Pas égal à (<>)  « ■ Différent de (<>) » (page 21-155) Vrai si N1 n'est pas égal à N2 ($N1 <> N2$). |
| | Supérieur à (>)  « ■ Supérieur à (>) » (page 21-155) Vrai si N1 est supérieur à N2 ($N1 > N2$). |
| | Supérieur ou égal à (>=)  « ■ Supérieur ou égal à (>=) » (page 21-155) Vrai si N1 est supérieur ou égal à N2 ($N1 >= N2$). |
| | Equivalent (==)  « ■ Egal à (==) » (page 21-155) Vrai si N1 est égal à N2 ($N1 = N2$). |

■ ET logique (and)

ET les côtés droit et gauche. La valeur 0 (zéro) est considérée comme désactivée et les autres valeurs comme activées.

N1 et N2 : Vrai si N1 et N2 sont ACTIVES. Sinon, faux.

■ OU logique (or)

OU les côtés droit et gauche. La valeur 0 (zéro) est considérée comme désactivée et les autres valeurs comme activées.

N1 ou N2 : Vrai si N1 ou N2 sont ACTIVES. Sinon, faux.

■ Négation (not)

Inverse la valeur. La valeur 0 (zéro) est considérée comme 1 et les autres valeurs comme 0.
 not N1 : Devient 0 si N1 est 1, et 1 si N1 est 0.

■ **inférieur à (<)**

Compare les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante.

Vrai si N1 est supérieur à N2 ($N1 < N2$).

■ **Inférieur ou égal à (<=)**

Compare les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante.

Vrai si N1 est inférieur ou égal à N2 ($N1 <= N2$).

■ **Différent de (<>)**

Compare les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante. Vrai si N1 n'est pas égal à N2 ($N1 <> N2$).

■ **Supérieur à (>)**

Compare les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante.

Vrai si N1 est supérieur à N2 ($N1 > N2$).

■ **Supérieur ou égal à (>=)**

Compare les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante.

Vrai si N1 est supérieur ou égal à N2 ($N1 >= N2$).

■ **Egal à (==)**

Compare les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante.

Vrai si N1 est égal à N2 ($N1 = N2$).

| Commande | | Par exemple |
|---------------------|-----|----------------------------------|
| ET logique | and | if ((Opération) and (Opération)) |
| OU logique | or | if ((Opération) or (Opération)) |
| Négation | not | if (not (Opération)) |
| Inférieur à | < | (Terme 1) < (Terme 2) |
| Inférieur ou égal à | <= | (Terme 1) <= (Terme 2) |
| Pas égal à | <> | (Terme 1) <> (Terme 2) |
| Supérieur à | > | (Terme 1) > (Terme 2) |
| Supérieur ou égal à | >= | (Terme 1) >= (Terme 2) |
| Egal à | == | (Terme 1) == (Terme 2) |

21.11.10 Opérateur

| Opérateur | Résumé de la fonction |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Addition (+) ☞ « ■ Addition (+) » (page 21-157) Ajoute les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante. |
| | Soustraction (-) ☞ « ■ Soustraction (-) » (page 21-157) Soustrait les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante. |
| | Module (%) ☞ « ■ Module (%) » (page 21-157) Détecte le reste d'une division effectuée sur les données dans deux adresses de mot ou les données dans une adresse de mot et une constante. |
| | Multiplication (*) ☞ « ■ Multiplication (*) » (page 21-157) Multiplie les données de deux adresses de mots ou les données dans une adresse de mot et une constante. |
| Opérateur Addition (+) Soustraction (-) Module (%) Multiplication (*) Division (/) Affectation (=) Décalage gauche (<<) Décalage gauche (>>) Opérateur sur bit ET logique (&) Opérateur sur bit OU logique () Opérateur sur bit OU exclusif (^) Opérateur sur bit Complément à 1 (~) | Division (/) ☞ « ■ Division (/) » (page 21-157) Divise les données en deux adresses de mot, ou les données dans une adresse de mot, par une constante. |
| | Affectation (=) ☞ « ■ Affectation (=) » (page 21-157) Affecte la valeur du côté droit dans le côté gauche. |
| | Décalage à gauche (<<) ☞ « ■ Décalage gauche (<<) » (page 21-157) Déplace les données vers le côté gauche selon le nombre qui se trouve au côté droit. |
| | Décalage à droite (>>) ☞ « ■ Décalage droite (>>) » (page 21-158) Déplace les données vers le côté droit selon le nombre qui se trouve au côté droit. |
| | Opérateur de bit ET logique (&) ☞ « ■ Bitwise AND (&) » (page 21-158) Effectue l'opération ET logique des données entre les périphériques de mot, ou entre les données de périphérique de mot et une constante. |
| | Opérateur de bit OU logique () ☞ « ■ Bitwise OR () » (page 21-158) Effectue l'opération OU logique des données entre les périphériques de mot, ou entre les données de périphérique de mot et une constante. |

Suite

| Opérateur | Résumé de la fonction |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opérateur | Opérateur de bit OU exclusif (^)  « ■ Bitwise OU exclusif (^) » (page 21-158) Effectue l'opération OU exclusif des données entre les périphériques de mot, ou entre les données de périphérique de mot et une constante. |
| | Opérateur sur bit Complément à 1 (~)  « ■ Bitwise 1's Complement (~) » (page 21-158) Inverse les bits. |

■ Addition (+)

Ajoute les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante. Si les résultats de calcul dépassent la zone, les chiffres sont tronqués.

■ Soustraction (-)

Soustrait les données dans deux adresses de mot, ou les données dans une adresse de mot et dans une constante. Si les résultats de calcul dépassent la zone, les chiffres sont tronqués.

■ Module (%)

Détecte le reste d'une division effectuée sur les données dans deux adresses de mot ou les données dans une adresse de mot et une constante. Le résultat de l'opération peut dépendre sur les signes qui se trouvent aux côtés gauches et droits.

■ Multiplication (*)

Multiplie les données de deux adresses de mots ou les données dans une adresse de mot et une constante. Si les résultats de calcul dépassent la zone, les chiffres sont tronqués.

■ Division (/)

Divise les données en deux adresses de mot, ou les données dans une adresse de mot, par une constante. Les valeurs fractionnaires à la suite de l'opération sont tronquées. Si les résultats de calcul dépassent la zone, les chiffres sont tronqués.

■ Affectation (=)

Affecte la valeur du côté droit dans le côté gauche. Seules des adresses peuvent être précisées sur le côté gauche. Les adresses et les constantes peuvent être utilisées sur le côté droit. Si les résultats de calcul dépassent la zone, les chiffres sont tronqués.

■ Décalage gauche (<<)

Déplace les données vers le côté gauche selon le nombre qui se trouve au côté droit. Cette fonction ne prend en charge que les décalages logiques.

Ordre de priorité et d'associativité

Le tableau suivant indique l'ordre de priorité des opérateurs. Si deux opérateurs ou plus disposent du même ordre de priorité, suivez la direction indiquée par l'associativité.

| Priorité | Opérateur | Associativité |
|----------|-----------|---------------|
| Haut | () | → |
| | not ~ | <← |
| | * / % | → |
| | + - | → |
| | << >> | → |
| | < <= > >= | → |
| | == <> | → |
| | & ^ | → |
| | and or | → |
| | Bas | = |

21.11.11 Opération de texte

Vous ne pouvez utiliser des fonctions d'opération de texte que dans un script étendu.

| Opération de texte | Résumé de la fonction |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>Conversion de l'entier à du texte décimal  « ■ Conversion Texte à Entier décimal » (page 21-161) Cette fonction sert à convertir du texte décimal en entiers.</p> |
| | <p>Conversion de l'entier à du texte hexadécimal  « ■ Conversion Texte à Entier hexadécimal » (page 21-163) Cette fonction sert à convertir du texte hexadécimal en entiers.</p> |
| | <p>A partir de la variable interne vers le tampon de données  « ■ Variable interne vers le tampon de données » (page 21-165) Les données de la chaîne stockée dans la mémoire interne sont copiées dans le tampon de données.</p> |
| | <p>A partir du tampon de données vers la variable interne  « ■ Tampon de données vers la variable interne » (page 21-167) Les données de la chaîne stockée dans le tampon de données sont copiées dans la mémoire interne.</p> |
|  | <p>Statut  « ■ Statut d'erreur de l'opération de texte » (page 21-169) Stocke toute erreur qui s'est produite.</p> |
| | <p>Conversion de la chaîne décimale de valeur numérique  « ■ Conversion de la chaîne décimale de valeur numérique » (page 21-171) Cette fonction sert à convertir un entier en une chaîne décimale.</p> |
| | <p>Conversion de la chaîne hexadécimale de valeur numérique  « ■ Conversion de la chaîne hexadécimale de valeur numérique » (page 21-172) Cette fonction sert à convertir des données binaires en une chaîne hexadécimale.</p> |
| | <p>Copier les chaînes partielles  « ■ Texte partiel » (page 21-173) Les données sont récupérées à partir du décalage de la chaîne selon la longueur de la chaîne, et stockées dans un autre tampon de données.</p> |
| | <p>Paramètres de texte  « ■ Paramètres de texte » (page 21-174) Une chaîne fixe est stockée dans le tampon de données.</p> |
| | <p>Longueur de chaîne  « ■ Longueur du texte » (page 21-175) Obtient la longueur de la chaîne stockée.</p> |
| | <p>Concaténation de chaîne  « ■ Concaténation de chaîne » (page 21-176) Une chaîne de caractères ou un code caractère est concaténé avec le tampon de texte.</p> |

■ Conversion Texte à Entier décimal

| Elément | Description |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Cette fonction sert à convertir une chaîne décimale en entiers. Convertissez le texte décimal dans le paramètre 2 (Tampon de données Convertir-à-partir) en un entier, et stockez-le dans le paramètre 1(Adresse Convertir-à). |
| Format | <p>_decasc2bin ([Adresse Convertir-à], [Tampon de données Convertir-à-partir])</p>  <p>Paramètre 1 : Variable interne, Adresse temporaire Paramètre 2 : Tampon de données</p> |

Exemple d'expression 1 (lorsque la longueur des données est de 16 bits)

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

Le contenu de «databuf0» est le suivant :

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

Les données ci-dessus sont converties comme suit :

| | 16 bits |
|--------|---------|
| LS0100 | 1234 |

Exemple d'expression 2 (lorsque la longueur des données est de 32 bits)

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

Le contenu de «databuf0» est le suivant :

| | 8 bits | |
|-------------|--------|------|
| 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 |

Les données ci-dessus sont converties comme suit :

| | 32 bits |
|--------|----------|
| LS0100 | 12345678 |
| LS0102 | |

IMPORTANT

- Une erreur se produit lorsque la longueur de bit convertie est plus élevée que la longueur de bit de l'éditeur D-Script.

Par exemple, lorsque la longueur de bit du script est de 16 bits :

```
_strset (databuf0, "123456") // Lorsqu'une chaîne décimale à 6 chiffres
est configurée par mégarde
```

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

Lorsque l'expression ci-dessus est exécutée, l'erreur numéro 2 (erreur de conversion de chaîne) du statut d'erreur de chaîne [e: Toutefois, le bit retourne au début de la fonction principale lorsqu'une erreur se produit. Donc, vous ne pouvez pas faire référence à d'autres fonctions directement après l'exécution de `_decasc2bin`. (Si la commande arrive lorsqu'une fonction est en cours d'exécution, elle retourne à la ligne qui a inséré la fonction.)

- Une erreur se produit lors de la conversion d'une chaîne de données contenant des caractères autres que «0» à «9».

Par exemple, lorsque la longueur de bit du script est de 16 bits :

```
_strset (databuf0, "12AB") // Lorsqu'une chaîne non décimale est
configurée par mégarde
```

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

Lorsque l'expression ci-dessus est exécutée, l'erreur numéro 2 (erreur de conversion de chaîne) du statut d'erreur de chaîne [e: Toutefois, le bit retourne au début de la fonction principale lorsqu'une erreur se produit. Donc, vous ne pouvez pas faire référence à d'autres fonctions directement après l'exécution de `_decasc2bin`. (Si la commande arrive lorsqu'une fonction est en cours d'exécution, elle retourne à la ligne qui a inséré la fonction.)

- Le traitement est terminé lorsqu'une erreur se produit et retourne au début de la fonction principale. (Si la commande arrive lorsqu'une fonction est en cours d'exécution, elle retourne à la ligne qui a inséré la fonction.)

■ Conversion Texte à Entier hexadécimal

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Cette fonction convertit une chaîne hexadécimale en données binaires. Convertissez le texte hexadécimal dans le paramètre 2 (Tampon de données Convertir-à-partir) en un entier, et stockez-le dans le paramètre 1 (Adresse Convertir-à). |
| Format | <p><code>_hexasc2bin ([Adresse Convertir-à], [Tampon de données Convertir-à-partir])</code></p>  <p>Paramètre 1 : Variable interne, Adresse temporaire Paramètre 2 : Tampon de données</p> |

Exemple d'expression 1 (lorsque la longueur des données est de 16 bits)

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

Le contenu de «databuf0» est le suivant :

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

Les données ci-dessus sont converties comme suit :

| | 16 bits |
|--------|---------|
| LS0100 | 1234h |

Exemple d'expression 2 (lorsque la longueur des données est de 32 bits)

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

Le contenu de «databuf0» est le suivant :

| 8 bits | | |
|-------------|-----|------|
| 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 |

Les données ci-dessus sont converties comme suit :

| 32 bits | |
|---------|-----------|
| LS0100 | 12345678h |
| LS0102 | |

IMPORTANT

- Une erreur se produit lorsque la chaîne convertie est supérieure à 16 octets ou 32 octets.

Par exemple, lorsque la longueur de bit du script est de 16 bits :

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

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

Lorsque l'expression ci-dessus est exécutée, l'erreur numéro 2 (erreur de conversion de chaîne) du statut d'erreur de chaîne [e:

- Une erreur se produit lors de la conversion d'une chaîne de données contenant des caractères autres que de «0» à «9», de «A» à «F», ou de «a» à «f».

Par exemple, lorsque la longueur de bit du script est de 16 bits :

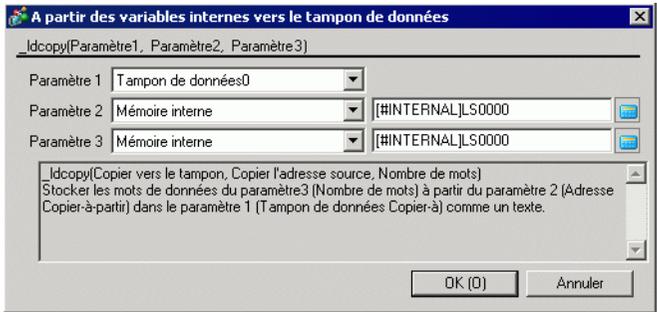
```
_strset (databuf 0, "123G")
```

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

Lorsque l'expression ci-dessus est exécutée, l'erreur numéro 2 (erreur de conversion de chaîne) du statut d'erreur de chaîne [e:

- Le traitement est terminé lorsqu'une erreur se produit et retourne au début de la fonction principale. (Si la commande arrive lorsqu'une fonction est en cours d'exécution, elle retourne à la ligne qui a inséré la fonction.)

■ Variable interne vers le tampon de données

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Les données de la chaîne stockées dans la zone LS sont copiées vers le tampon de données selon le nombre de chaînes dans un transfert octet par octet. Stockez les mots du paramètre 3 (nombre de mots) du paramètre 2 (adresse copier à partir de) dans le paramètre 1 (tampon des données copier vers) comme texte. |
| Format | <p><code>_Idcopy (Tampon de données Copier vers, [Adresse Copier à partir], Nombre de mots)</code></p>  <p>Paramètre 1 : Tampon de données Paramètre 2 : Variable interne Paramètre 3 : Valeur entière, Variable interne, Adresse temporaire (la plage valide pour le paramètre 3 est entre 1 et 1024.)</p> |

Exemple d'expression 1 :

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

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

Les données de LS0100 à LS0103 sont écrites séquentiellement dans les 4 octets du tampon des données, à compter de «databuf0». La zone LS est lue dans chaque octet (les octets les plus bas).

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

IMPORTANT

- L'octet 1 bas de la zone LS est lu et la quantité précisée de données est écrite dans le tampon des données.
- La valeur maximale que vous pouvez affecter pour le paramètre 3 est de 1024. Lorsque vous configurez une valeur qui dépasse la limite, le numéro d'erreur 1 (débordement de chaîne) du statut d'erreur de chaîne [e: STR_ERR_STAT] est déclenchée.
- Même s'il y a des données dans l'octet supérieur de la variable interne, seules les données dans l'octet inférieur sont lues.
- Le traitement est terminé lorsqu'une erreur se produit et retourne au début de la fonction principale. (Si la commande arrive lorsqu'une fonction est en cours d'exécution, elle retourne à la ligne qui a inséré la fonction.)

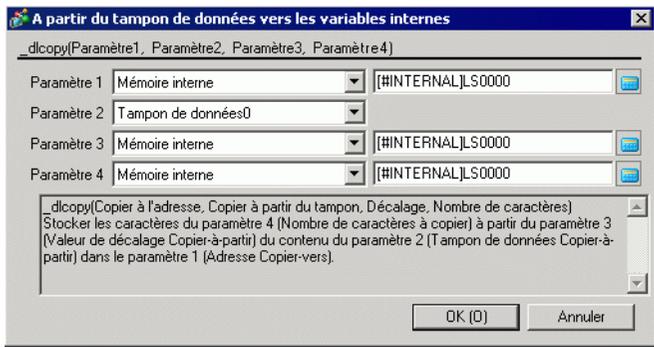
`_ldcopy (databuf0, [w:[#INTERNAL]LS0100], 4)`

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

Lorsque les données sont stockées comme dans l'illustration plus haute, les données du dernier octet sont lues et écrites dans le tampon de données.

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

■ Tampon de données vers la variable interne

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Chaque octet de données de chaîne stocké dans le décalage du tampon de données est copié vers la zone LS selon le nombre de chaînes. Stocke les caractères de données du paramètre 4 (Nombre de caractères à copier) à partir du paramètre 3 (Valeur de décalage Copier-à-partir) du contenu du paramètre 2 (Tampon de données Copier-à-partir) dans le paramètre 1 (Adresse Copier-vers). |
| Format | <p><code>_dlcopy</code> ([Adresse Copier vers], Tampon de données Copier à partir, Valeur de décalage Copier à partir, Nombre de caractères copiés)</p>  <p>Paramètre 1 : Variable interne Paramètre 2 : Tampon de données Paramètre 3 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 3 est entre 0 et 1024.) Paramètre 4 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 4 est entre 1 et 1024.)</p> |

Exemple d'expression 1 :

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

| | 8 bits | |
|-------------|--------|-----|
| 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 octets de données récupérés à partir de «offset 2» de «databuf0» sont écrits dans LS0100 et LS0103. Les données sont écrites dans la zone LS en unités de 1 octet.

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

IMPORTANT

- 1 octet de données est lu à partir du tampon de données et écrit dans la zone LS. Cela signifie que seuls les derniers 8 bits (1 octet) dans la zone LS seront utilisés. Les 8 bits (1 octet) significatifs seront effacés.
 - Lorsque la valeur spécifiée [valeur du décalage source + nombre de caractères à copier] est plus élevée que la taille du tampon de données, l'erreur numéro 3 (erreur d'extraction de chaîne) du statut d'erreur de chaîne [e: STR_ERR_STAT] est émise.
 - Le traitement est terminé lorsqu'une erreur se produit et retourne au début de la fonction principale. (Si la commande arrive lorsqu'une fonction est en cours d'exécution, elle retourne à la ligne qui a inséré la fonction.)
-

■ Statut d'erreur de l'opération de texte

Lorsqu'une erreur se produit lors de l'exécution d'une opération de texte, une erreur est configurée au statut d'erreur de l'opération de texte [e: STR_ERR_STAT]. «0» dans [e: STR_ERR_STAT] indique une condition normale, et des valeurs autres que «0» stockées dans [e: STR_ERR_STAT] indiquent les états des erreurs. L'erreur la plus récente est stockée dans le statut d'erreur de l'opération de texte [e: STR_ERR_STAT]. Le statut d'erreur de l'opération de texte peut être configuré dans les paramètres d'étiquette de l'[Opération de port SIO] dans le menu Boîte à outils D-Script. Le tableau suivant donne les erreurs d'opération de texte.

| Numéro d'erreur | Message d'erreur | Description |
|-----------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | Normal | Pas d'erreur |
| 1 | Débordement de texte | <p>Une chaîne contenant au moins 256 octets est directement incluse dans l'argument pour les fonctions suivantes : <code>_strset ()</code>, <code>_strlen ()</code>, <code>_strcat ()</code>, <code>_strmid ()</code>, et <code>IO_READ_WAIT ()</code>.</p> <p>Ou, une chaîne qui dépasse la taille du tampon de données lors de l'exécution des fonctions <code>_strcat ()</code> ou <code>_ldcopy ()</code>.</p> <p>Par exemple :</p> <pre>_strcat (databuf0, databuf1)</pre> <p>La fonction ci-dessus est exécutée lorsqu'une chaîne de 1020 octets est stockée dans <code>databuf0</code>, et qu'une chaîne de 60 octets est stockée dans <code>databuf 1</code>. (Une chaîne qui dépasse 1024 octets, ce qui est la taille du tampon de données, produit un statut d'erreur.)</p> |
| 2 | Erreur de conversion de chaîne | <p>Un code de caractère non valide est donné à la fonction <code>_hexasc2bin ()</code> ou <code>_decasc2bin ()</code>.</p> <p>Par exemple :</p> <p>Un code de caractère autre que de «0» à «9», de «A» à «F», ou de «a» à «f» est inclus dans le deuxième argument de <code>_hexasc2bin ()</code>.</p> |
| 3 | Erreur de récupération de chaîne | <p>La récupération d'une chaîne de caractères qui est plus longue que la chaîne de caractères précisée dans la fonction «<code>_strmid ()</code>» est tentée. Ou, une valeur de décalage qui dépasse la chaîne précisée est désignée.</p> <p>Par exemple :</p> <pre>_strmid (databuf0, "12345678", 2, 8)</pre> <p>La récupération d'une chaîne à 8 caractères à partir du décalage 2 est tentée.</p> |

Vous ne pouvez pas utiliser le statut d'erreur de contrôle de chaîne avec des D-scripts ou des D-scripts globaux. S'il est lu par mégarde, la valeur «0» sera chargée.

Il est stocké dans le statut d'erreur lors de l'exécution de chaque fonction.

Pour vérifier l'erreur [e: STR_ERR_STAT], écrivez les instructions suivantes. Vous pouvez confirmer l'erreur à l'aide de l'expression suivante.

Exemple d'expression :

```
if ([e:STR_ERR_STAT] <> 0)           // Vérifie le statut d'erreur.
{
    set ([b:[#INTERNAL]LS005000])    // Configure un bit sur le voyant d'affichage
d'erreur
}
endif
```

IMPORTANT

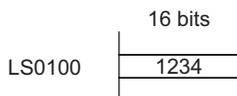
- Le traitement est terminé lorsqu'une erreur se produit et retourne au début de la fonction principale. (Si la commande arrive lorsqu'une fonction est en cours d'exécution, elle retourne à la ligne qui a inséré la fonction.)
-

■ Conversion de la chaîne décimale de valeur numérique

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Cette fonction sert à convertir un entier en une chaîne décimale. Convertissez l'entier dans le paramètre 2 (Adresse Convertir-à-partir) en un texte décimal, et stockez-le dans le paramètre 1(Tampon de données Convertir-à). |
| Format | <p><code>_bin2decasc(Adresse du résultat de la conversion, Tampon de source de la conversion)</code></p>  <p>Paramètre 1 : Tampon de données Paramètre 2 : Variable interne, Adresse temporaire</p> |

Exemple d'expression 1 (lorsque la longueur des données est de 16 bits)

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

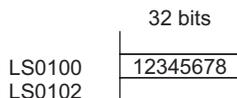


Les données ci-dessus sont converties comme suit : Remarquez que «NULL (0x00)» est ajouté.

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

Exemple d'expression 2 (lorsque la longueur des données est de 32 bits)

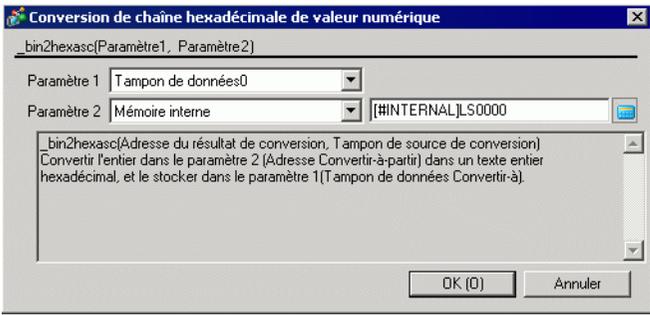
`_bin2decasc (databuf0, [w:[#INTERNAL]LS0100])`



Les données ci-dessus sont converties comme suit :

| | | |
|-------------|--------|------|
| | 8 bits | |
| 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 |

■ Conversion de la chaîne hexadécimale de valeur numérique

| Elément | Description |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Cette fonction sert à convertir des données binaires en une chaîne hexadécimale. Convertissez l'entier dans le paramètre 2 (Adresse Convertir-à-partir) en un texte hexadécimal, et stockez-le dans le paramètre 1 (Tampon de données Convertir-à). |
| Format | <p><code>_bin2hexasc (Tampon de données Convertir-à, [Adresse Convertir-à-partir])</code></p>  <p>Paramètre 1 : Tampon de données Paramètre 2 : Variable interne, Adresse temporaire</p> |

Exemple d'expression 1 (lorsque la longueur des données est de 16 bits)

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

| | 16 bits |
|--------|---------|
| LS0100 | 1234h |

Les données ci-dessus sont converties comme suit : Remarquez que «NULL (0x00)» est ajouté.

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

Exemple d'expression 2 (lorsque la longueur des données est de 32 bits)

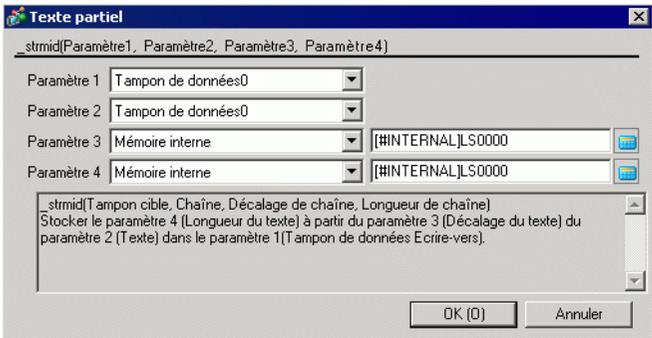
`_bin2hexasc (databuf0, [w:[#INTERNAL]LS0100])`

| | 32 bits |
|--------|-----------|
| LS0100 | 12345678h |
| LS0102 | |

Les données ci-dessus sont converties comme suit :

| | 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 |

■ **Texte partiel**

| Elément | Description |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Les données sont récupérées à partir du décalage de la chaîne selon la longueur de la chaîne, et stockées dans un autre tampon de données. Stockez le paramètre 4 (Longueur du texte) depuis le paramètre 3 (décalage du texte) du paramètre 2 (Texte) dans le paramètre 1 (Tampon de données Ecrire dans). |
| Format | <p>_strmid (Tampon de données Ecrire vers, Texte, Décalage du texte, Longueur du texte)</p>  <p>Paramètre 1 : Tampon de données Paramètre 2 : Chaîne, Tampon de données Paramètre 3 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 3 est de 0 à 1024.) Paramètre 4 : Valeur numérique, Variable interne, Adresse temporaire (la plage valide pour le paramètre 4 est de 1 à 1024.)</p> |

Exemple d'expression :

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

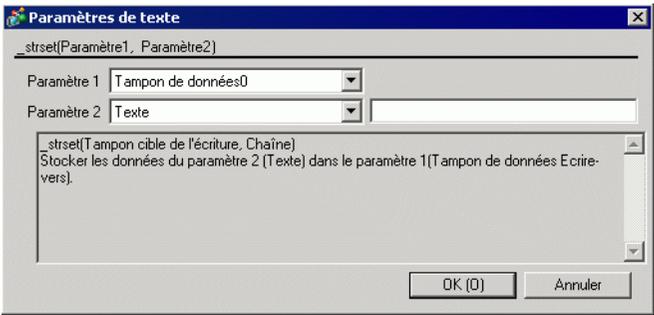
4 octets de données récupérées à partir du décalage 2 de la chaîne «12345678» sont stockés dans «databuf0».

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

IMPORTANT

- Lorsque vous tentez de récupérer une chaîne qui est plus longue que la chaîne précisée dans la fonction «strmid ()», ou lorsque vous précisez une valeur de décalage qui dépasse la chaîne précisée, l'erreur numéro 3 (erreur d'extraction de chaîne) du statut d'erreur de chaîne [e: STR_ERR_STAT] est émise.
- Le traitement est terminé lorsqu'une erreur se produit et retourne au début de la fonction principale. (Si la commande arrive lorsqu'une fonction est en cours d'exécution, elle retourne à la ligne qui a inséré la fonction.)

■ Paramètres de texte

| Elément | Description |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Une chaîne fixe est stockée dans le tampon de données. Stocke les données figurant dans le paramètre 2 (Texte) dans le paramètre 1 (Tampon de données Ecrire dans). |
| Format | <p><code>_strset(Tampon de destination de l'écriture, Chaîne)</code></p>  <p>Paramètre 1 : Tampon de données Paramètre 2 : Texte, Valeur numérique (Code texte) (la plage valide pour le paramètre 2 est de 0 et de 1 à 255.)</p> |

Exemple d'expression :

`_strset (databuf0, "ABCD")`

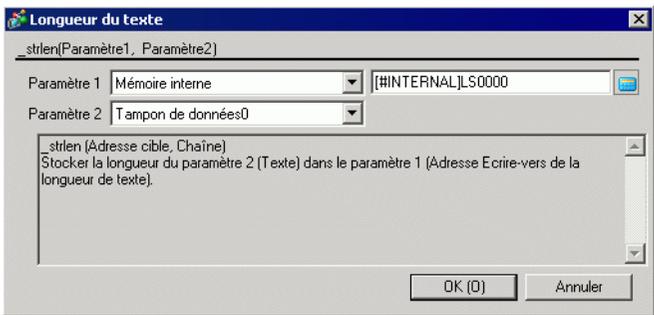
Dans l'illustration suivante, la chaîne est stockée dans le tampon de données :

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

IMPORTANT

- Vous pouvez préciser une chaîne contenant jusqu'à 255 caractères. Pour créer des chaînes qui dépassent la limite, stockez la chaîne dans un autre tampon et concaténez les chaînes à l'aide de la fonction de concaténation de chaînes (`_strcat`).
- Pour effacer le tampon de données, créez une chaîne vide `""`.
 Par exemple : `_strset (databuf0, "")`
`_strset (databuf0, 0)`

■ Longueur du texte

| Elément | Description |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Obtient la longueur de la chaîne stockée. Stocke la longueur du paramètre 2 (Texte) dans le paramètre 1 (Adresse Ecrire dans de la longueur du texte). (Le caractère NULL n'est pas inclus.) |
| Format | <p><code>_strlen (Adresse cible, Chaîne)</code></p>  <p>Paramètre 1 : Variable interne, Adresse temporaire Paramètre 2 : Chaîne, Tampon de données</p> |

Exemple d'expression 1 :

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

Lorsque l'instruction ci-dessus est exécutée, la longueur de la chaîne est écrite dans LS0100 tel qu'illustré ci-dessous.



Exemple d'expression 2 :

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

Le contenu de «databuf0» est le suivant :

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

Lorsque l'instruction ci-dessus est exécutée, la longueur de la chaîne est écrite dans [t: 0000] tel qu'illustré ci-dessous.



■ Concaténation de chaîne

| Elément | Description |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Résumé | Une chaîne de caractères ou un code caractère est concaténé avec le tampon de texte. Ajoute les données figurant dans le paramètre 2 (Texte) au contenu du paramètre 1 (Tampon de données de contact). |
| Format | <p><code>_strcat(Tampon de données de chaîne, Chaîne)</code></p>  <p>Paramètre 1 : Tampon de données Paramètre 2 : Texte, Valeur numérique (Code texte), Tampon de données (la plage valide pour le paramètre 2 est de 0 et de 1 à 255.)</p> |

Exemple d'expression 1 :

`_strcat (databuf0, "ABCD")`

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

Lorsque la chaîne «ABCD» est concaténée selon la configuration ci-dessus, le résultat est le suivant. Remarquez que «NULL (0x00)» est ajouté.

| | 8 bits | |
|-------------|--------|------|
| 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 |

IMPORTANT

- Vous pouvez préciser une chaîne contenant jusqu'à 255 caractères.
- Si vous définissez une chaîne vide "" pour la valeur numérique 0 dans le paramètre 2, le tampon de données du paramètre 1 ne change pas. Par exemple : `_strset (databuf0, "")`
`_strset (databuf0, 0)`

21.11.12 Exemple d'opération

■ Exemples d'opérations logiques

Le tableau suivant montre quelques exemples des opérations logiques :

◆ **((100 > 99) and (200 <> 100))**

Résultat : Activé

◆ **((100 > 99) and (200 <> 200))**

Résultat : Désactivé

◆ **((100 > 99) or (200 <> 200))**

Résultat : Activé

◆ **((100 < 99) or (200 <> 200))**

Résultat : Désactivé

◆ **not (100 > 99)**

Résultat : Désactivé

◆ **not (100 < 99)**

Résultat : Activé

◆ **[w:[PLC1]D200] < 10**

Résultat : Vrai si D200 est inférieur à 10.

◆ **not [w:[PLC1]D200]**

Résultat : Vrai si D200 est égal à 0.

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

Résultat : Vrai si D200 est égal à 2 ou à 5.

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

Résultat : Vrai si D200 est inférieur à 5, et si D300 est inférieur à 8.

◆ **[w:[PLC1]D200] < 10**

Résultat : Vrai si D200 est inférieur à 10.

◆ **not [w:[PLC1]D200]**

Résultat : Vrai si D200 est égal à 0.

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

Résultat : Vrai si D200 est égal à 2 ou à 5.

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

Résultat : Vrai si D200 est inférieur à 5, et si D300 est inférieur à 8.

■ Exemples d'opérations sur bit

Le tableau suivant montre quelques exemples des opérations sur bit :

◆ [w:[PLC1]D200] << 4

Résultat : Les données figurant dans D200 sont déplacées 4 bits vers la gauche.

◆ [w:[PLC1]D200] >> 4

Résultat : Les données figurant dans D200 sont déplacées 4 bits vers la droite.

◆ 12(0000Ch) est stocké dans D301, en format BIN.

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

Résultat : Les données figurant dans D300 sont déplacées 12 bits vers la gauche et affectées à D200.

◆ [w:[PLC1]D200] << 4

Résultat : Les données figurant dans D200 sont déplacées 4 bits vers la gauche.

◆ [w:[PLC1]D200] >> 4

Résultat : Les données figurant dans D200 sont déplacées 4 bits vers la droite.

◆ 12(0000Ch) est stocké dans D310, en format BIN.

[w:[PLC1]D200] = [w:[PLC1]D300] >> [w:[PLC1]D310]

Résultat : Les données figurant dans D300 sont déplacées 12 bits vers la gauche et affectées à D200.

◆ Bitwise AND

| | |
|-----------------|-------------------|
| 0 & 0 | Résultat : 0 |
| 0 & 1 | Résultat : 0 |
| 1 & 1 | Résultat : 1 |
| 0x1234 & 0xF0F0 | Résultat : 0x1030 |

◆ Bitwise OR

| | |
|-----------------|-------------------|
| 0 0 | Résultat : 0 |
| 0 1 | Résultat : 1 |
| 1 1 | Résultat : 1 |
| 0x1234 0x9999 | Résultat : 0x9BBD |

◆ Bitwise XOR

| | |
|-------|--------------|
| 0 ^ 0 | Résultat : 0 |
| 0 ^ 1 | Résultat : 1 |
| 1 ^ 1 | Résultat : 0 |

◆ Bitwise 1's Complement (lorsque le format des données est BIN16+)

| | |
|-----|-------------------|
| ~ 0 | Résultat : 0xFFFF |
| ~ 1 | Résultat : 0xFFFE |

■ Exemples de calculs de l'usage du branchement conditionnel

Contrôlez le déroulement du programme à l'aide des commandes «if-endif» et «if-else-endif».

◆ if-endif

```
if (condition)
{Process 1}
endif
```

Si la condition est vraie, le processus 1 est exécuté. Si non, le processus 1 est ignoré.

Par exemple :

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

Si les données figurant dans D200 sont inférieures à 5, 1 est affecté à D100.

◆ if - else - endif

```
if (condition)
{Process 1}
else
{Process 2}
endif
```

Si la condition est vraie, le processus 1 est exécuté. Sinon, le processus 2 est exécuté.

Par exemple :

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

Si la valeur dans D200 est inférieure à 5, 1 est affecté à D100. Sinon, 0 est affecté.

■ Exemples de calculs d'usage de l'adresse de décalage

Spécification du décalage : Exemples de calculs spéciaux à l'aide de [w:D00100]#[t:0000].

- ◆ Script E/S : 16 bits non signés, [t:0000]= 65526, l'adresse résultante est [w:[PLC1]D00090].

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

Les derniers 16 bits sont valides

- ◆ Script E/S : 16 bits signés, [t:0000]= -10, l'adresse résultante est [w:[PLC1]D00090].

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

Les derniers 16 bits sont valides

- ◆ Script E/S : 32 bits non signés, [t:0000]= 4294901840, l'adresse résultante est [w:[PLC1]D00180].

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

Les derniers 16 bits sont valides

- ◆ Script E/S : 32 bits signés, [t:0000]= -65456, l'adresse résultante est [w:[PLC1]D00180].

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

Les derniers 16 bits sont valides

IMPORTANT

- Les adresses de décalage sont toujours traitées en tant que valeurs Bin 16 bits, peu importe la longueur de bit du script et les paramètres de type de données. Si le résultat dépasse 16 bits (valeur maximale : 65535), les bits 0 à 15 sont traités en tant que bits valides et les bits 16 et supérieurs sont supprimés.