

SIEMENS

SINUMERIK Operate
Principes fondamentaux du langage
de haut niveau

SINUMERIK
828D/840D sl

Édition 03/2018
Manuel de formation

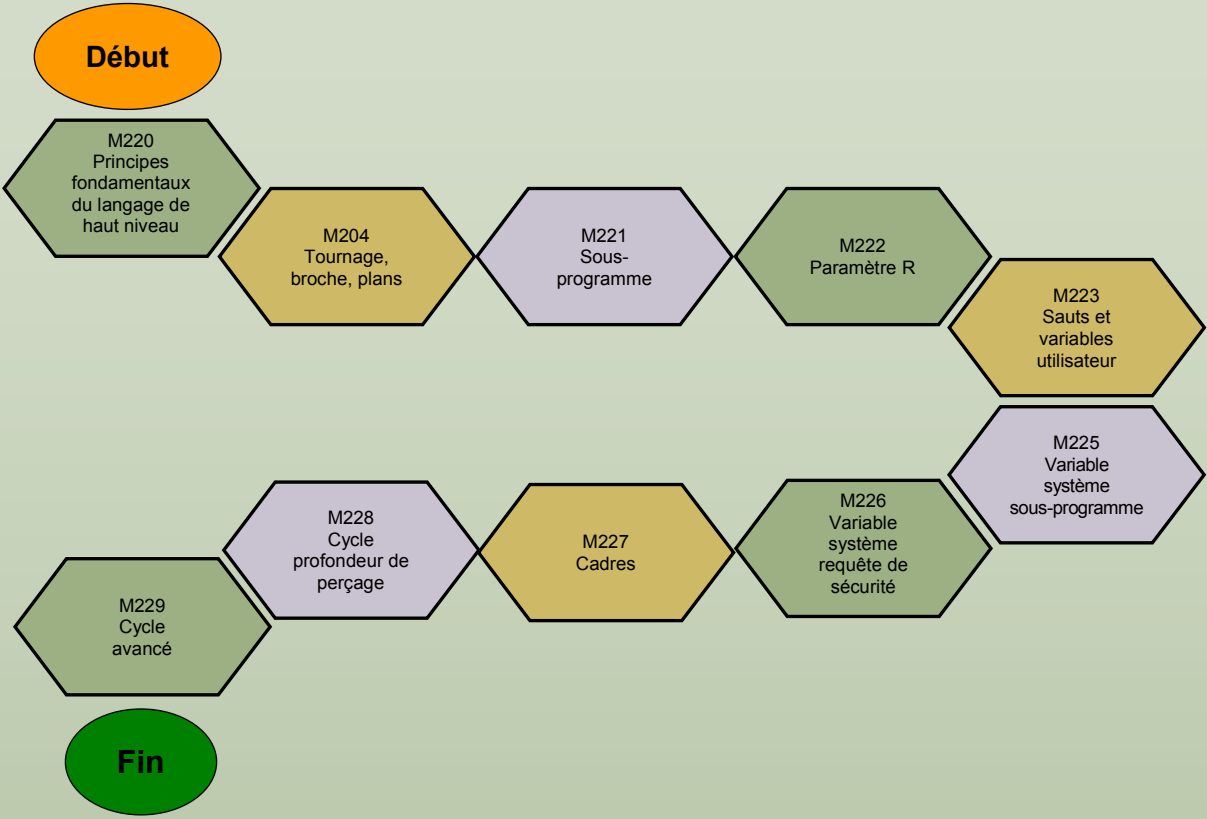
SINUMERIK

Principes fondamentaux du langage de haut niveau

Valable pour :

SINUMERIK 828D	SW4.7
SINUMERIK 840D sl	SW4.7

Contenu



Description du module :

La procédure du sujet "Principes fondamentaux du langage de haut niveau" est décrite dans ce module. Le terme "Langage de haut niveau" est défini et décrit pour SINUMERIK.

La structure générale des modules "Principes fondamentaux du langage de haut niveau" est décrite ici.

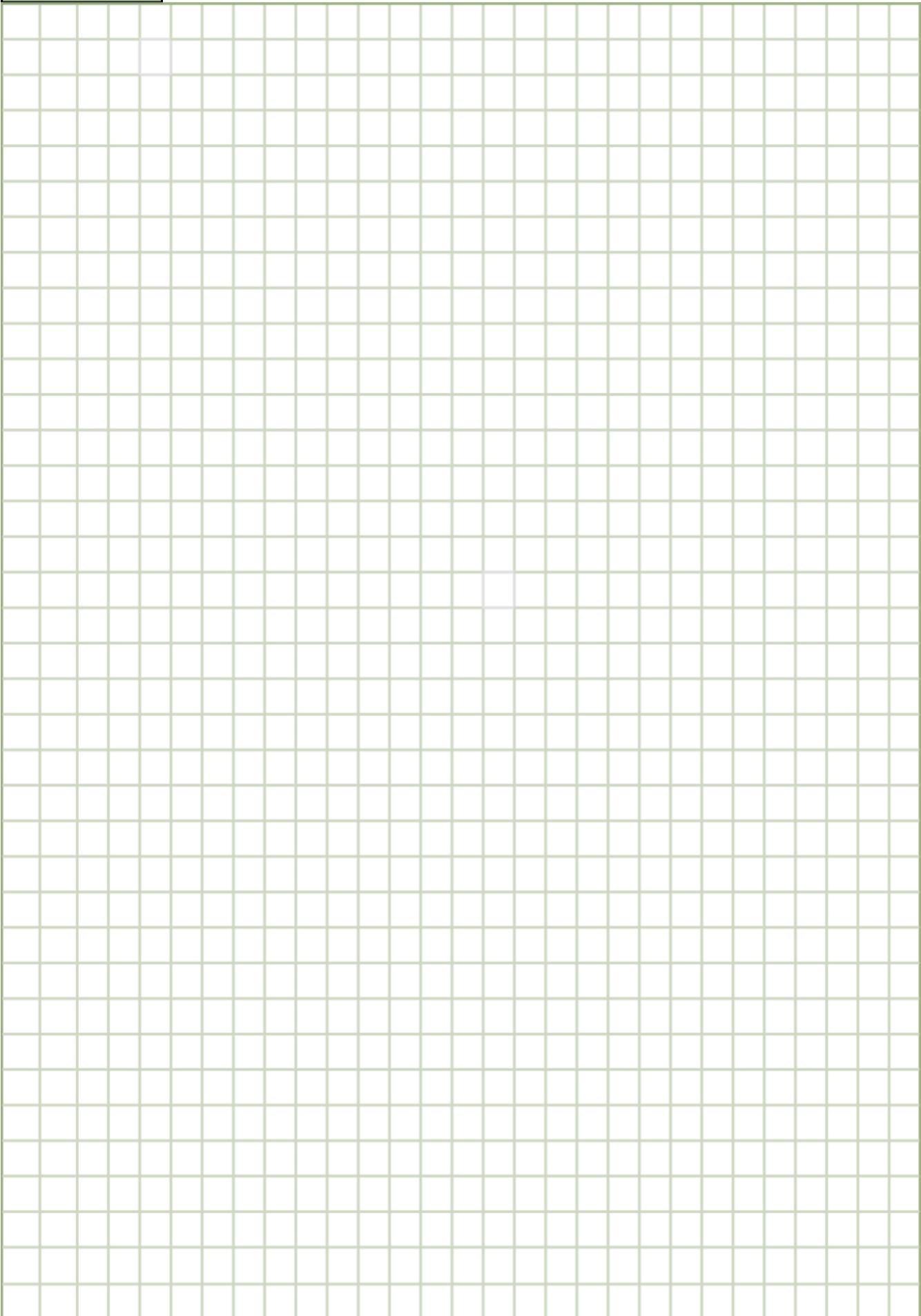
Cas d'utilisation :

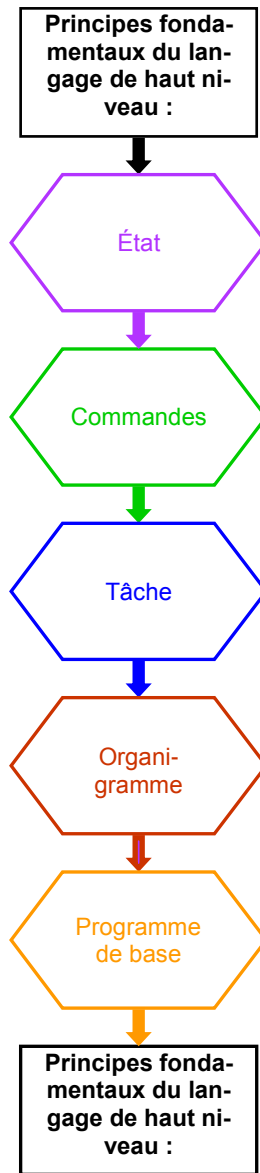
Vous apprendrez comment appliquer les "Principes fondamentaux du langage de haut niveau", de façon à pouvoir facilement utiliser et mettre en œuvre les éléments et les exemples fournis dans votre application.

Contenu :

Définition du "Langage de haut niveau" en lien avec SINUMERIK

Mise en œuvre du contenu de la documentation "Principes fondamentaux du langage de haut niveau"

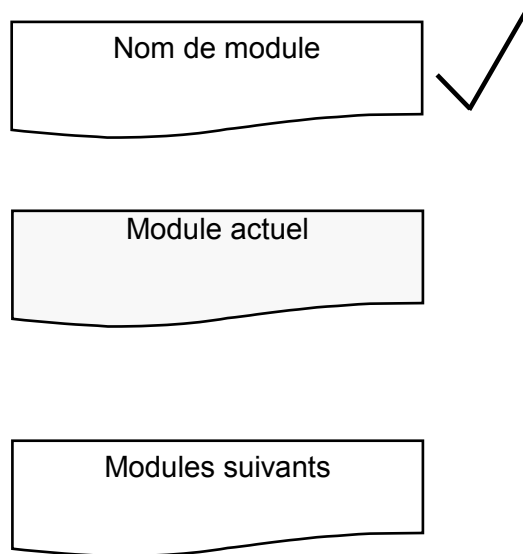




Remarques

État :

Au début de chaque module, un graphique affiche l'état, c'est-à-dire les sujets qui ont été traités, les sujets en cours de traitement et les sujets qui seront traités dans les modules suivants.



Il s'agit de programmes CNC préconfigurés pour les modules "Principes fondamentaux du langage de haut niveau".

Ces programmes CNC sont clairement assignés aux modules correspondants par nom. Ils contiennent une solution suggérée pour les tâches des modules.

De cette façon, il existe toujours un programme exécutable pour chaque module.

Il est recommandé de consulter la documentation complémentaire pour les modules existants :

Disponible sur l'extranet TAC.

Module :
Langage de haut niveau DE800D_Complete

Disponible sur Internet.

Manuel de la liste des variables système SINUMERIK.
Planification de la production pour SINUMERIK.
Principes fondamentaux pour le manuel de programmation SINUMERIK.

Disponible sur DOC ON CD ou sur Internet sous la forme d'un PDF à télécharger.
Saisir simplement les termes ci-dessus dans un moteur de recherche Internet.

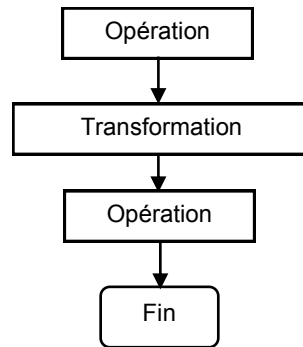
Remarques

Il existe une distinction entre deux sujets :

la programmation CNC axée sur la machine.

Il s'agit de commandes du langage de haut niveau utilisées pour la programmation linéaire.

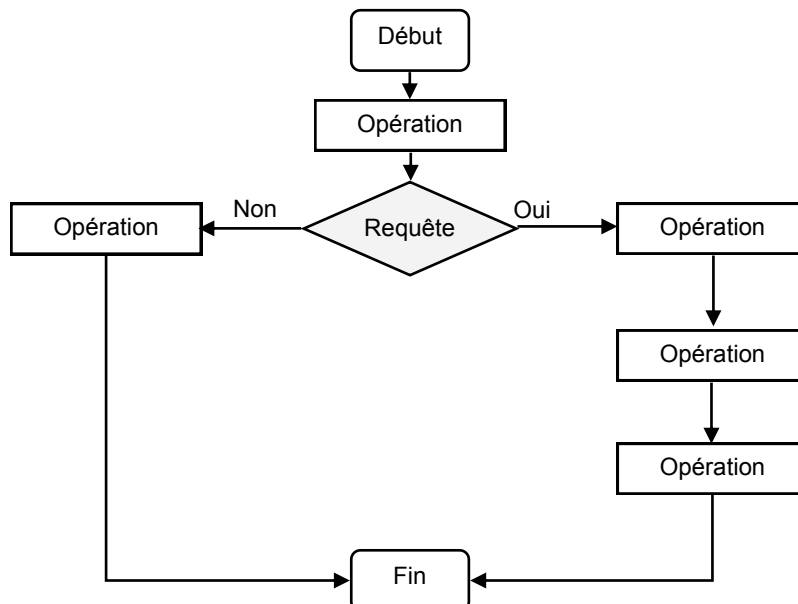
Programmation CNC axée sur la machine



Le langage de programmation CNC de plus haut niveau :

Il s'agit de commandes du langage de haut niveau utilisées pour la programmation procédurale axée sur les fonctions.

Langage de programmation CNC de plus haut niveau



Remarques

Commandes :

Étape 1 :

À la première étape, les commandes utilisées dans le programme sont répertoriées avec une brève description.

Exemple :

Général

<u>Commande</u>	<u>Signification</u>
Commande 1	Description
Commande 2	Description
Commande 3	Description
Commande 4	Description

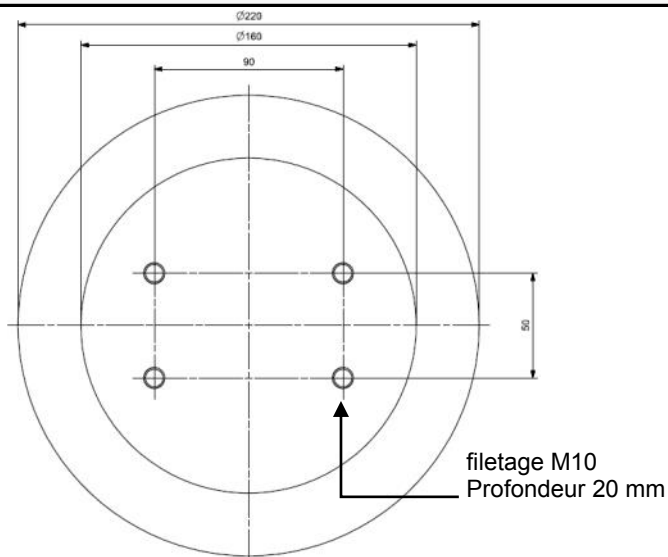
Tâche :

Étape 2 :

Description de la tâche :

La tâche avec les conditions requises du programme est décrite dans cette fenêtre.

Exemple :



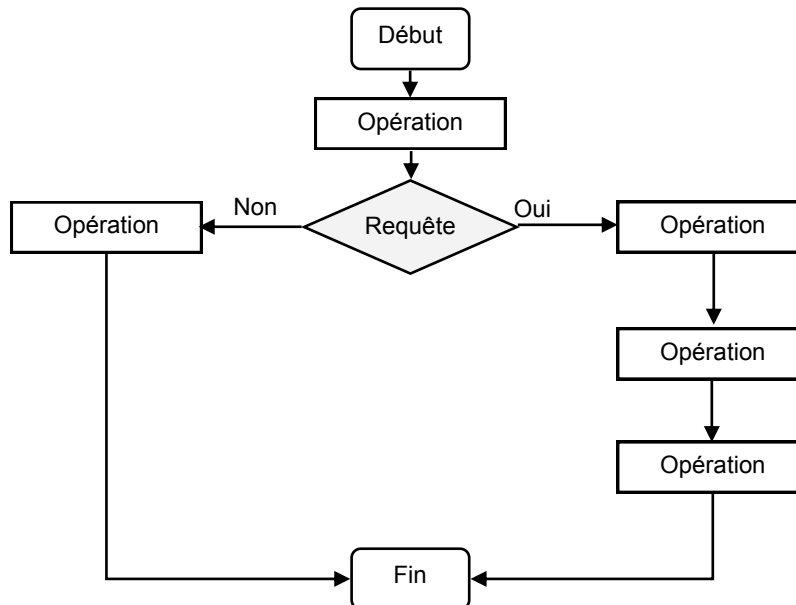
Remarques

État :

Étape 3 :

Un organigramme est créé pour la tâche.

Exemple :



Étape 4 :

Le programme est créé.

Exemple :

Cycle

```

PROC DRILL (REAL _STARTP, REAL _RETRAC, REAL _DEPTH_TOTAL, REAL _DEPTH_1, REAL _FEED, REAL
_FEEED_1, REAL _DEPTH_MAX, REAL _DEPTH_MIN) ;DISPLOF SAVE
;-----
DEF REAL _DEPTH_CAL
DEF REAL _LAST_STEP
DEF INT _STEP
DEF REAL _STEP_EV
DEF REAL _ACTPOS
DEF INT _TOOLTYPE
;-----

_DEPTH_TOTAL=ABS(_DEPTH_TOTAL)
_DEPTH_1=ABS(_DEPTH_1)

IF _FEED==0 GOTOF _ERROR2
ENDIF

_TOOLTYPE=$TC_DP1[$P_TOOLNO,1]
IF (_TOOLTYPE<200) OR (_TOOLTYPE>299) GOTOF _ERROR1

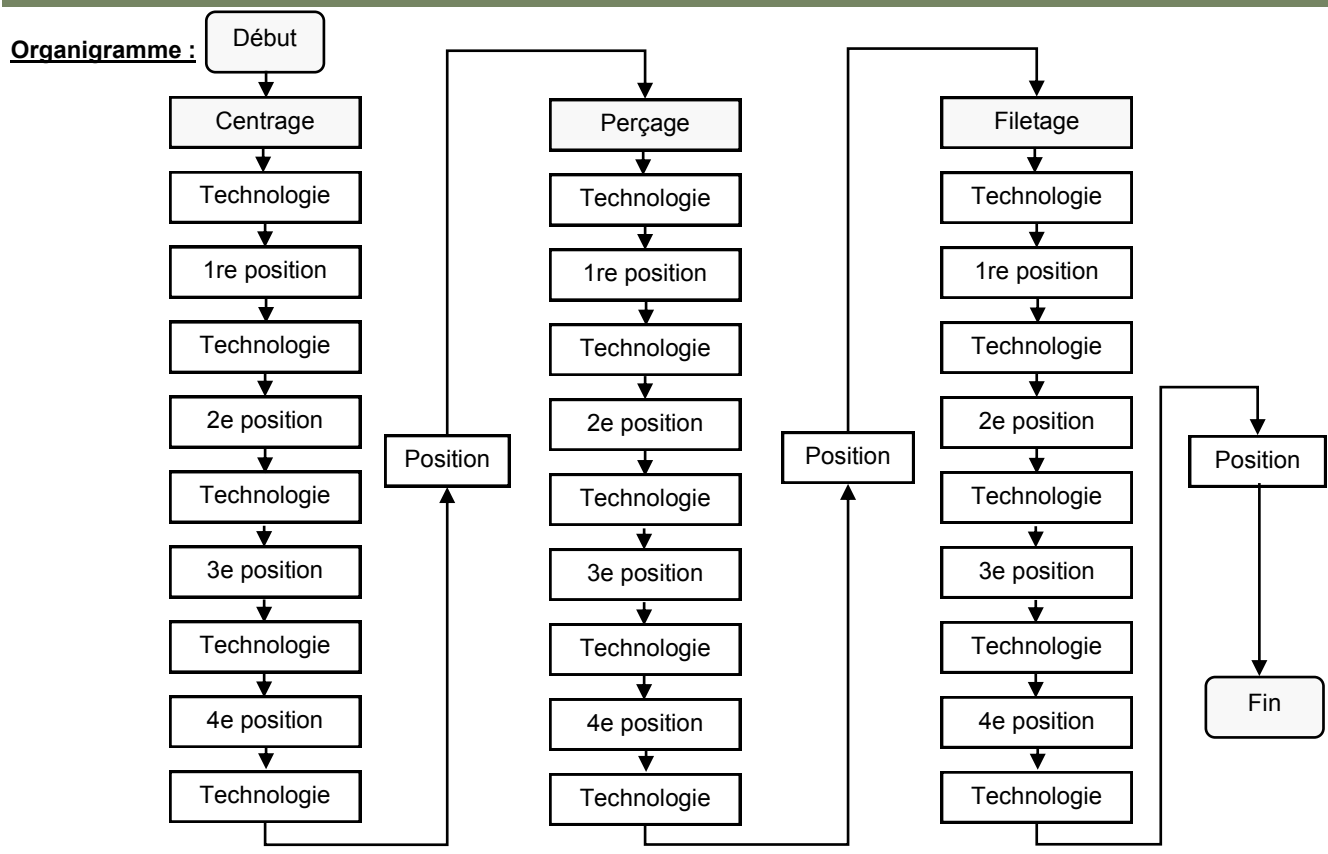
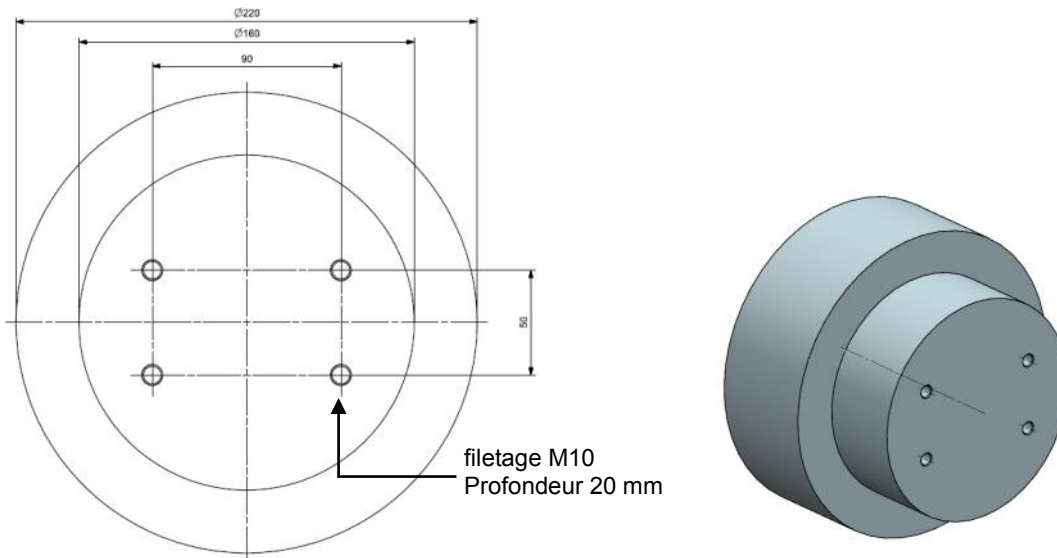
```

Chaque module dispose du programme approprié sous la forme d'un "fichier MPF". Ce programme peut être immédiatement chargé et simulé.

Remarques

Le programme de base est fondé sur la documentation existante : Former le formateur, tournage. Dans ce programme, il n'existe aucun cycle ni aucune commande de langage de haut niveau. Ils seront ajoutés au fur et à mesure.

Schéma :



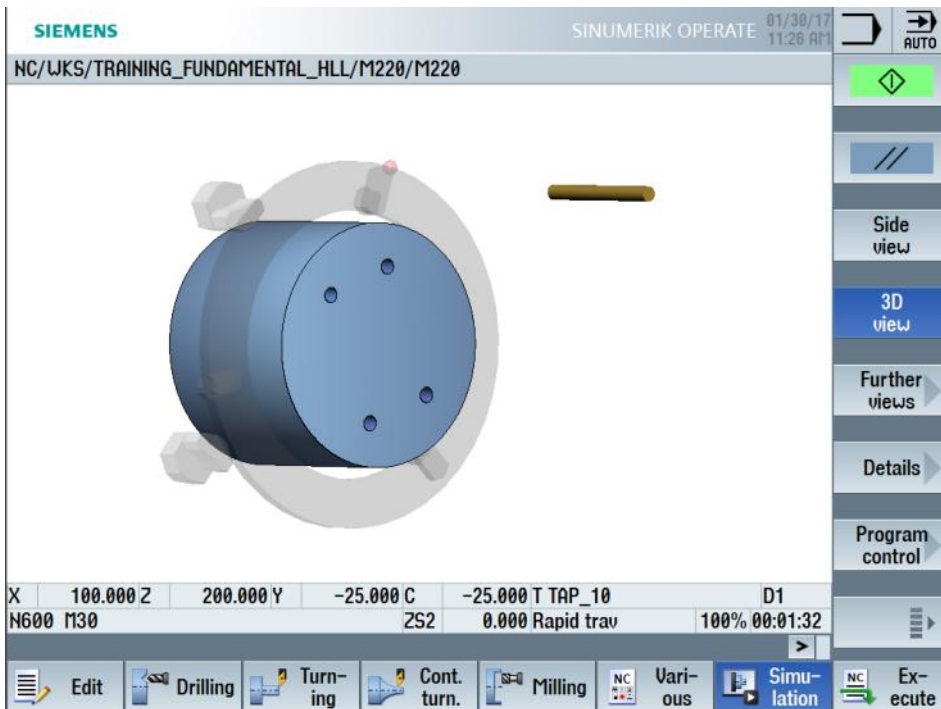
Remarques

Programme de base :

```
N10 G17 G60 G90 G94
N20 G54
N30 WORKPIECE(,,, "CYLINDER",192,0,-120,-100,160)
N40 G18 ;TURNINGPLANE
N50 SPOS=0 ;SPINDLEPOSITIONING
N60 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N70 T="CENTERDRILL"
N80 S3=1000 M3=3
N90 TRANSMIT ;TRANSFORM MILLING INTO TURNING
N100 DIAMOF ;PROGRAMMING IN RADIUS
N110 G17 ;MILLINGPLANE
N120 F140
N130 G0 Z3
N140 G0 X-45 Y25
N150 G1 Z-2
N160 G0 Z3
N170 G0 X45 Y25
N180 G1 z-2
N190 G0 Z3
N200 G0 X45 Y-25
N210 G1 z-2
N220 G0 Z3
N230 G0 X-45 Y-25
N240 G1 z-2
N250 G0 Z3
N260 G0 X100 Z200
N270 T="DRILL_8_5"
N280 M3=3
N290 F180
```

```
N300 G0 Z3
N310 G0 X-45 Y25
N320 G1 Z-22
N330 G0 Z3
N340 G0 X45 Y25
N350 G1 z-22
N360 G0 Z3
N370 G0 X45 Y-25
N380 G1 z-22
N390 G0 Z3
N400 G0 X-45 Y-25
N410 G1 z-22
N420 G0 Z3
N430 G0 X100 Z200
N440 T="TAP_10" ;TAPPING TOOL
N450 S200 M3=3
N460 G0 Z3
N470 G0 X-45 Y25
N480 G331 Z-20 K1.5 S200
N490 G332 Z3 K1.5 S400
N500 G0 X45 Y25
N510 G331 Z-20 K1.5 S200
N520 G332 Z3 K1.5 S400
N530 G0 X45 Y-25
N540 G331 Z-20 K1.5 S200
N550 G332 Z3 K1.5 S400
N560 G0 X-45 Y-25
N570 G331 Z-20 K1.5 S200
N580 G332 Z3 K1.5 S400
N590 G0 X100 Z200
N600 M30
```

Remarques

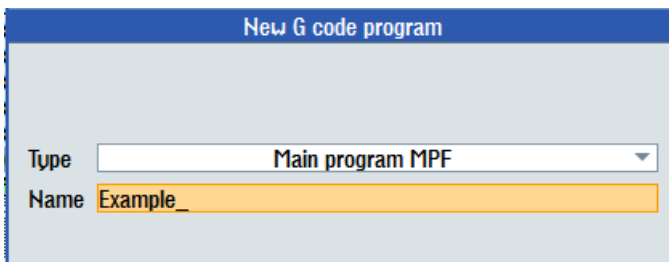


Le programme de base est terminé.

Ce programme est désormais optimisé avec des commandes de langage de haut niveau dans le module suivant.

Il est recommandé de copier ce programme dans SinuTrain de façon à pouvoir suivre les modules suivants.

⇒ Créer un nouveau programme en code G.



⇒ Copier le programme en code G depuis ce module dans l'éditeur du programme en code G nouvellement créé.

Le programme de base pour les modules suivants est alors disponible et il est donc possible de suivre facilement les étapes suivantes.

Remarques

Description du module :

Le travail avec différentes broches et affectations, ainsi que la modification des plans de travail, sont décrits dans ce module.

Cas d'utilisation :

Après avoir terminé ce module, vous pourrez configurer des broches dans les programmes CNC en fonction de l'usinage requis.

Les opérations d'usinage pourront donc être réalisées plus rapidement.

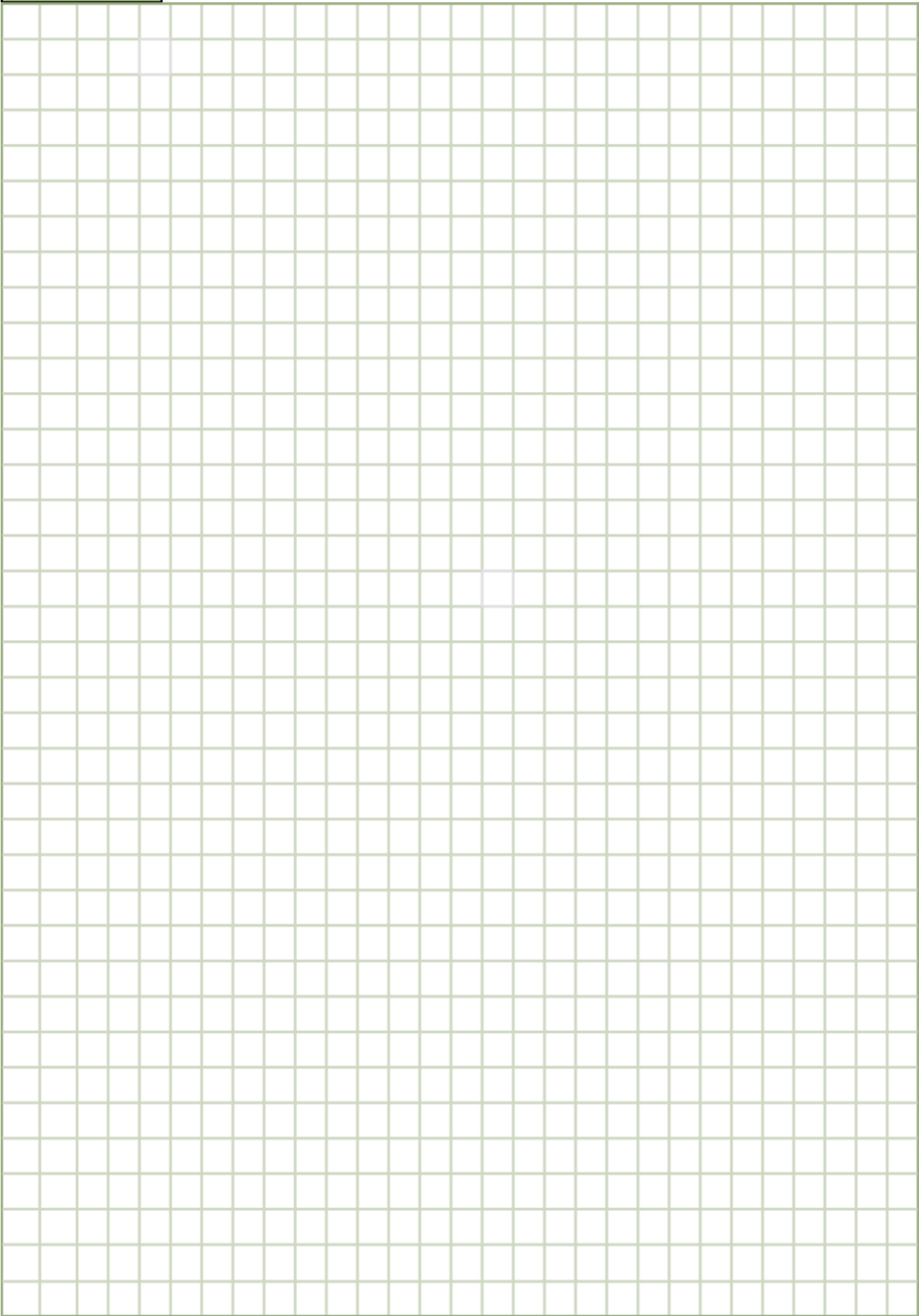
Exemple : Différents processus d'usinage dans différents plans sur une machine CNC.

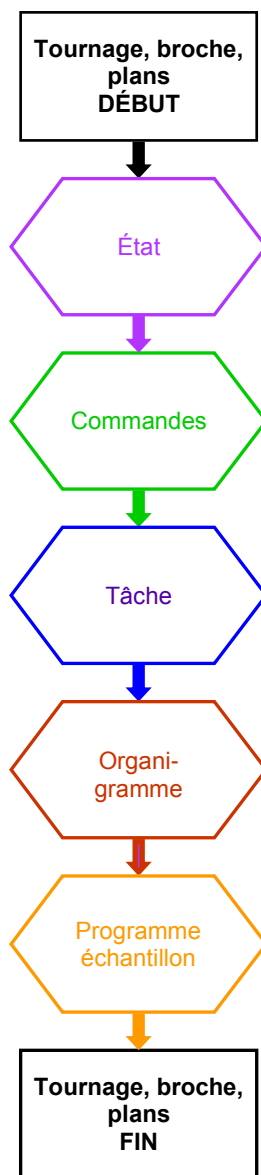
Contenu :

Explication des commandes requises du langage de haut niveau

Explication des broches et de la sélection de broche

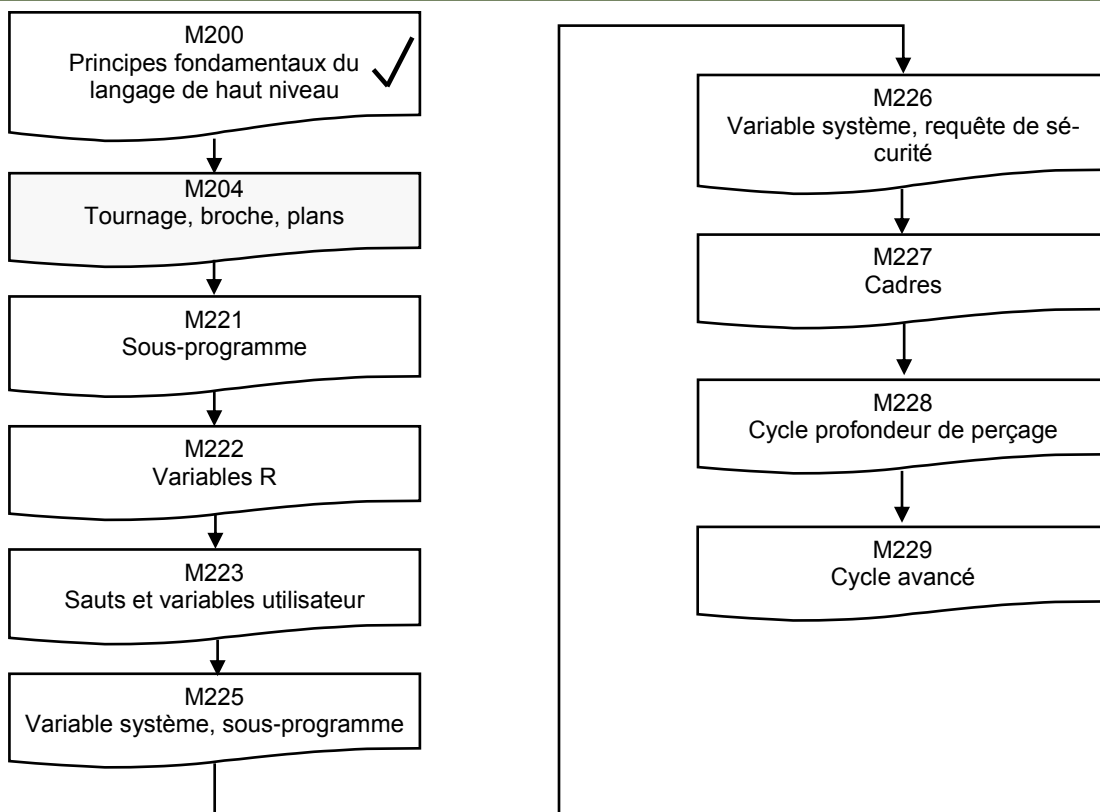
Explication des plans et de leur sélection





Remarques

État :

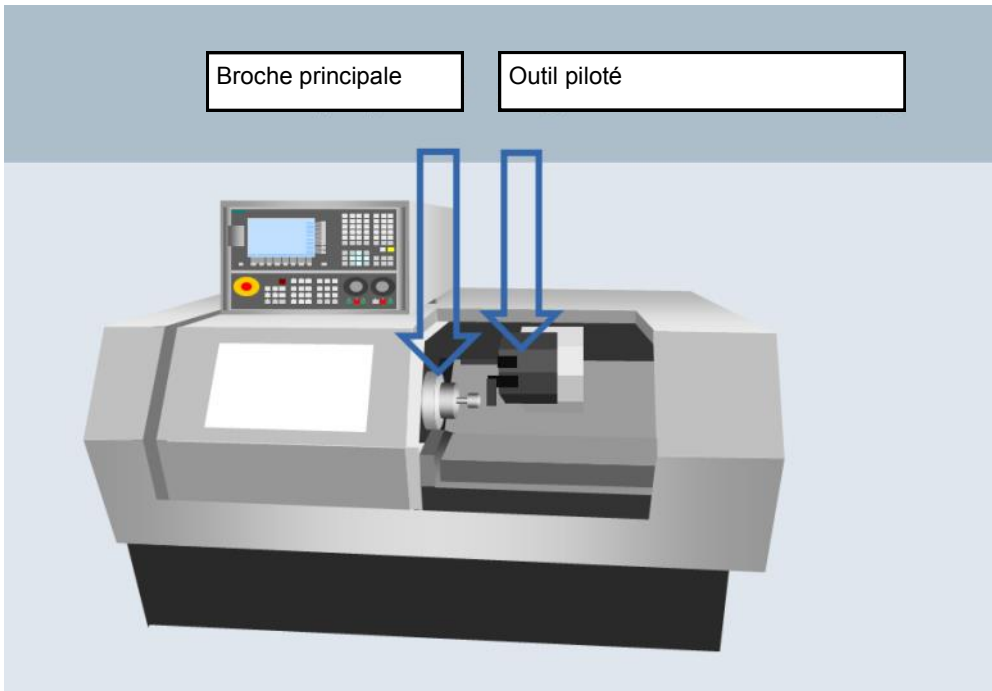


Commandes :

<u>Commande</u>	<u>Signification</u>
SETMS	Retour à la broche maîtresse définie
SETMS(n)	Définition d'une broche arbitraire comme broche maîtresse
SPOS	Positionnement de la broche maîtresse
DIAMOF	Programmation en fonction du rayon
TRANSMIT	Fonction pour la transformation de la face finale
DIAMON	Programmation en fonction du diamètre
TRAFOOF	Les transformations actives sont désactivées

Remarques

Ce module se réfère à un tour avec la configuration d'axe suivante :



SETMS

Cette commande permet de revenir à la broche maîtresse.

La cinématique d'une machine CNC est définie par le fabricant de la machine.

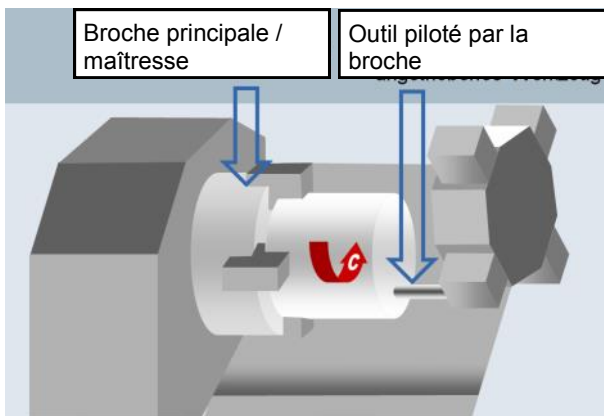
Quel que soit le nombre de broches dont dispose la machine, il n'existe toujours qu'une seule broche maîtresse, dont dépendent toutes les autres broches.

Le fabricant définit généralement la broche maîtresse via les données de la machine.

Ces données de la machine ne font pas partie de ce module.

Généralement, la broche principale de la machine est définie comme broche maîtresse, mais ce n'est pas toujours le cas.

Exemple d'un tour :



Remarques

Les commandes If SETM... sont souvent utilisées dans les programmes CNC et il est recommandé d'écrire la commande "SETMS" au début d'un programme.

Le programmeur s'assure ainsi que la broche maîtresse du fabricant de la machine est également active.

Important : La vitesse S et l'avance F dans le programme se réfèrent toujours à la broche maîtresse.

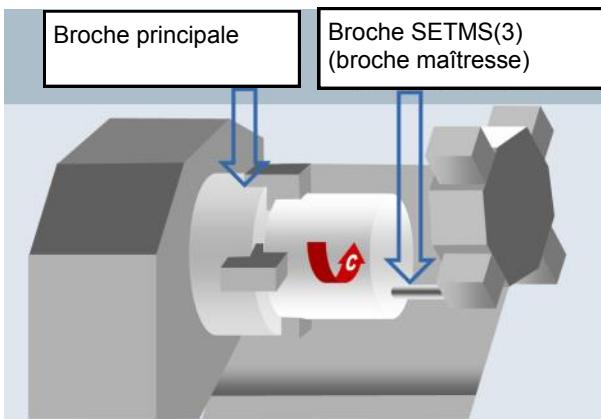
SETMS (n)

Définition d'une broche comme broche maîtresse.

En pratique, il peut être nécessaire de définir une autre broche de la machine comme broche maîtresse. Par exemple, les données techniques peuvent être affectées à cette broche.

Il est également important de connaître la désignation de la broche (numéro), fournie par le fabricant de la machine dans ce cas.

Exemple pour SETMS (3)



La broche principale du tour n'est plus désormais la broche maîtresse, mais plutôt la broche de l'outil piloté. Par conséquent, toutes les valeurs technologiques se réfèrent à cette broche.

Remarques

SPOS :

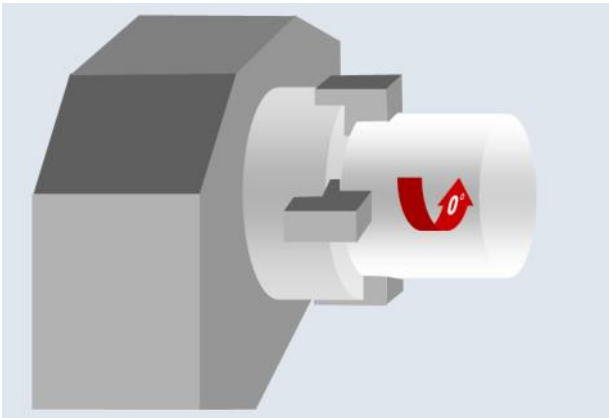
La broche maîtresse active peut être positionnée à un certain nombre de degrés avec la commande SPOS.

Exemple : La broche principale est également la broche maîtresse.

SPOS=0

La broche maîtresse est positionnée à 0 degré, est en régulation de position et a basculé en mode axe.
Si la vitesse de la broche (S) est programmée, le mode broche change automatiquement.

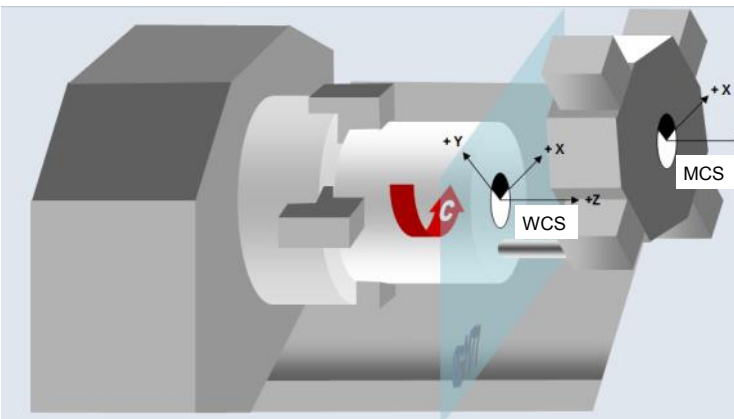
SPCOF désactive la régulation de position.

**DIAMOF :**

Programmation en fonction du rayon.

TRANSMIT :

La commande TRANSMIT (**Trans**form **M**illing into **T**urning) effectue la transformation des axes de fraisage et de perçage avec les outils pilotés sur la face finale. Le plan G17 doit d'abord être sélectionné.



Les déplacements suivants peuvent être effectués dans le système de coordonnées cartésiennes (X,Y), normalement utilisé en fraisage. Le contrôle convertit les blocs de programme pour les axes réels (X, C). L'axe Z reste inchangé.

Remarques

DIAMON :

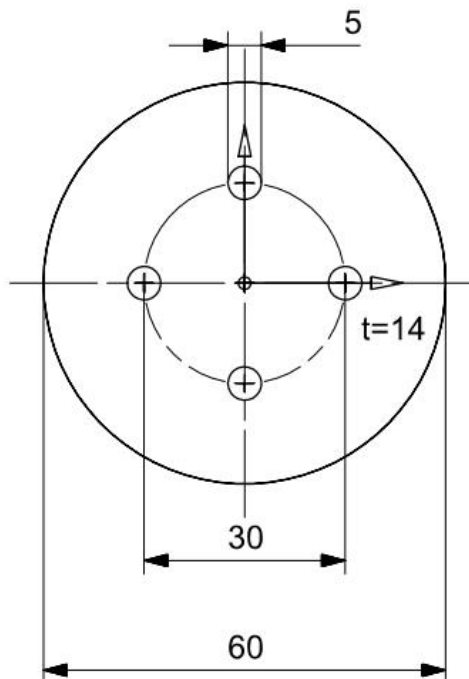
Programmation en fonction du diamètre. Typiques pour les tours.

TRAFOOF :

Les transformations actives sont désactivées. Cette commande doit être insérée à la fin du programme pour désactiver la fonction TRANSMIT.

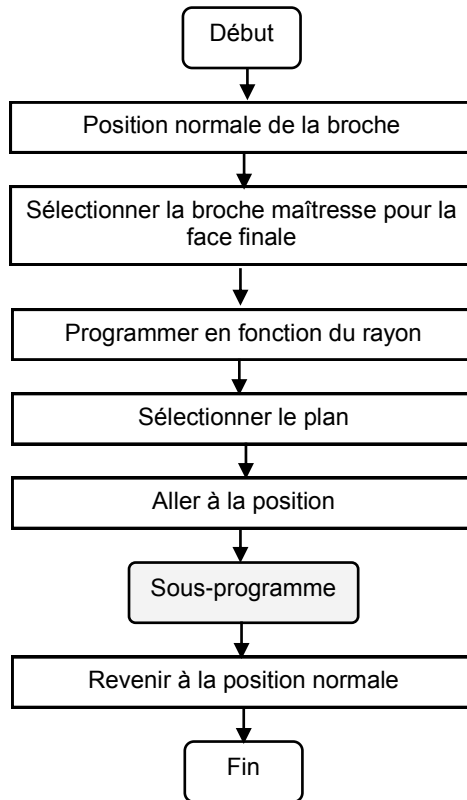
Tâche :

- ⇒ Les trous doivent être percés sur la face finale d'une pièce arrondie, avec un diamètre de 60 mm.
- ⇒ La broche de l'outil piloté doit être la broche maîtresse.
- ⇒ Le plan de perçage sur la face finale doit être actif.
- ⇒ La programmation des positions de perçage doit être effectuée en fonction du rayon.
- ⇒ Les positions de perçage doivent être écrites dans un sous-programme.
- ⇒ À la fin du programme, la machine doit revenir à sa position normale.

Schéma :

Remarques

Organigramme :

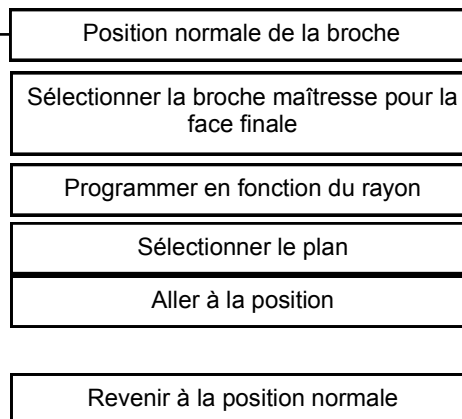


Programme échantillon :

Programme principal

```

N10 WORKPIECE(,,, "CYLINDER", 0, 0, -100, -80, 60)
N20 G54 G60 G90 G95
N30 SETMS
N40 G18
N50 SPOS=0
N60 T="DRILL_5" D1
N70 SETMS(3)
N80 S=1000 F0.2 M3
N90 DIAMOF
N100 G17
N110 G0 X15 Z2
N120 TRANSMIT
N130 DRILLING
N140 G0 Z100
N150 TRAFOOF
N160 DIAMON
N170 SETMS
N180 M30
  
```



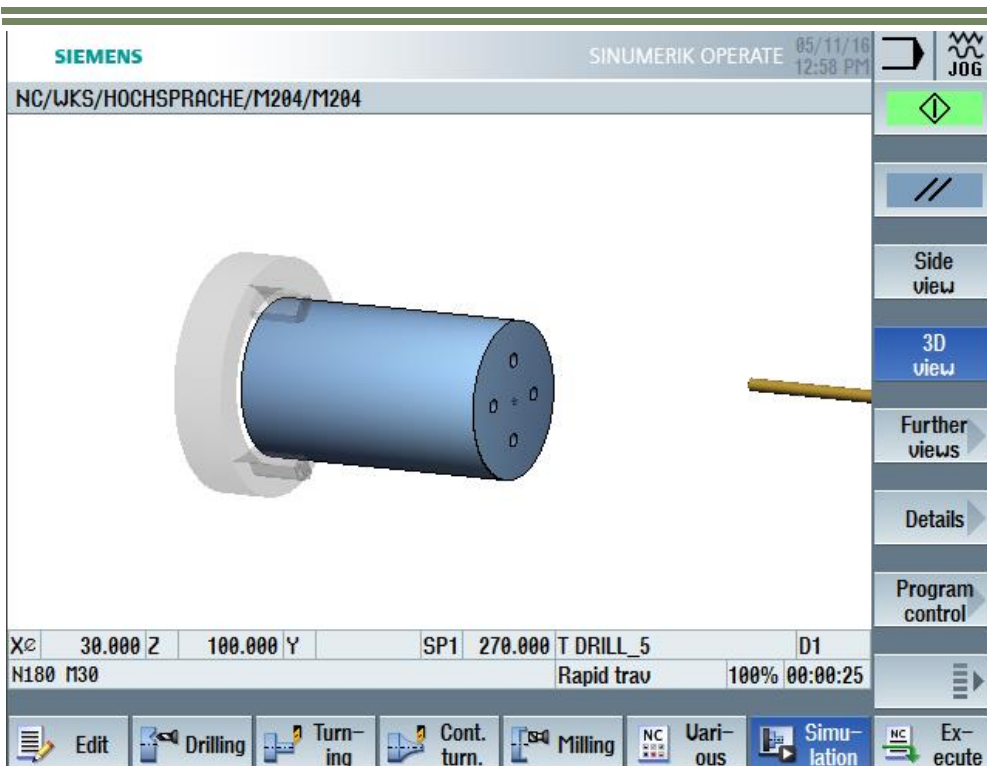
Remarques

Sous-programme :

```

N10 G1 Z-14
N20 G0 Z2
N30 G0 X0 Y15
N40 G1 Z-14
N50 G0 Z2
N60 G0 X-15 Y0
N70 G1 Z-14
N80 G0 Z2
N90 G0 X0 Y-15
N100 G1 Z-14
N110 G0 Z2
N120 M17

```



Ce module peut constituer la base de tous les autres sujets (modules) dans les principes fondamentaux du langage de haut niveau, si la formation des sujets se réfère à un tour.

Remarques

Description du module :

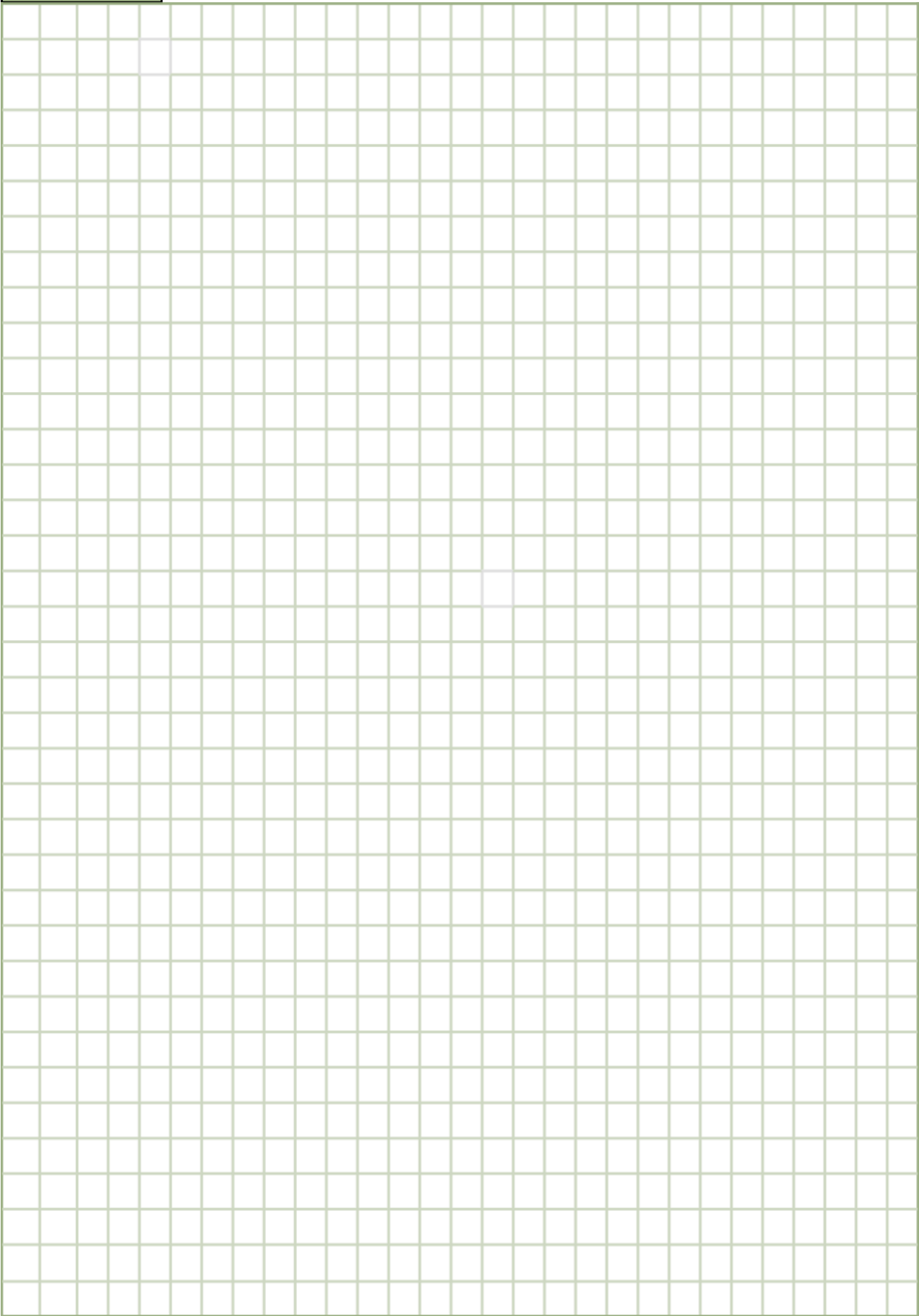
Ce module décrit la création de sous-programmes, basée sur un exemple.

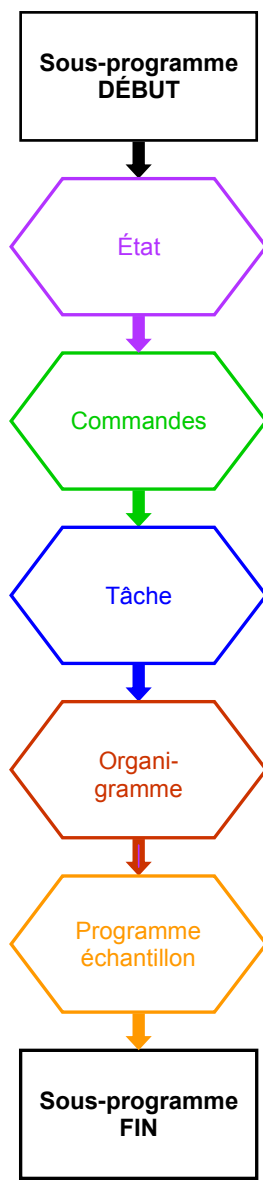
Cas d'utilisation :

Après avoir terminé ce module, vous pourrez utiliser des sous-programmes et, par conséquent, optimiser la structure et la configuration de vos programmes CNC. Cela vous permettra d'économiser du temps et les programmes CNC pourront être configurés de façon à faciliter leur lecture et leur usage par d'autres développeurs.

Explication des commandes pour travailler avec des sous-programmes

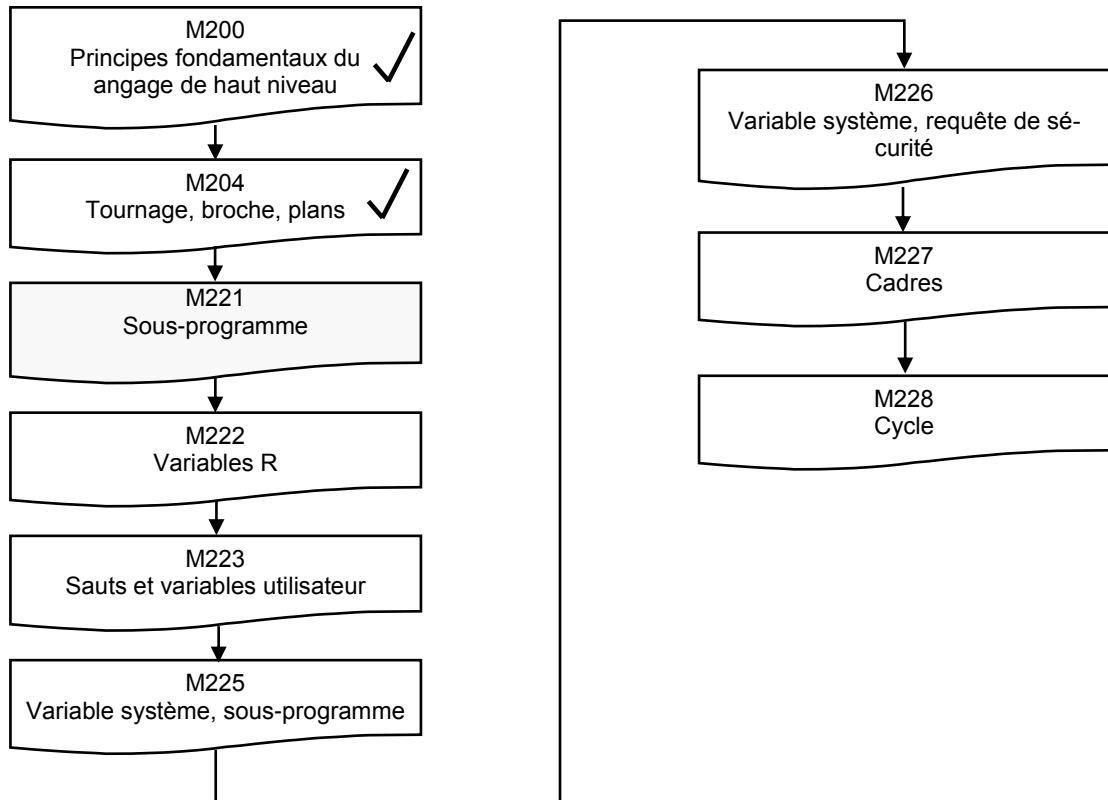
Implémentation des contenus, basée sur un exemple





Remarques

État :



Travail avec des sous-programmes :

- ⇒ Produit une structure de programme propre. Le programme est plus clair et plus facile à lire.
- ⇒ Les sections du programme peuvent être utilisées plusieurs fois, également en tant que "bloc" et bibliothèque d'usinage.
- ⇒ Comme il n'est plus nécessaire de réécrire à chaque fois les mêmes sections de programme, cela fait gagner du temps lors de la création d'un programme et économise de l'espace mémoire.

Structure d'un sous-programme :

En principe, le sous-programme est structuré de la même façon que le programme principal. Il existe des commandes spécifiques au sous-programme en fonction de l'application.

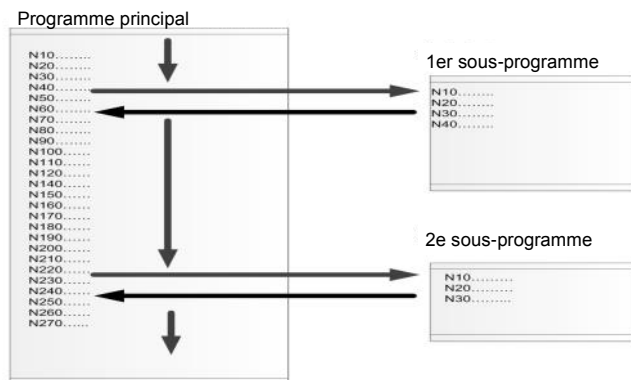
L'appel des sous-programmes est soumis à une certaine hiérarchie :

"Nomduprogramme.MPF" est trouvé lorsqu'il est dans le même répertoire (WPD).

"Nomduprogramme.SPF" est trouvé précisément selon l'ordre de recherche suivant :

1. Dans le répertoire actif
2. Dans le dossier "Subprograms"
3. Dans le répertoire des cycles utilisateur
4. Dans le répertoire des cycles fabricant
5. Dans le répertoire des cycles standard

Cette question est traitée en détail dans les modules suivants.



Remarques

Commandes :

<u>Commande</u>	<u>Signification</u>
M17	Fin du sous-programme
RET	Fin du sous-programme
Nom	Un nom arbitraire identifie un sous-programme

M17 : Marque la fin du sous-programme avec pour instruction de revenir au programme principal. Le retour s'effectue via l'API.

RET : Marque également la fin du sous-programme, mais sans interrompre le mode chemin continu et sans sortie fonctionnelle vers l'API. (Retour plus rapide que M17)

Nom : Les mêmes règles (conventions DOS) s'appliquent aux noms des sous-programmes et du programme principal : 24 caractères maximum pour le nom d'un programme. Ils s'affichent dans le NC.

⇒ Lettres A...Z et a...z

⇒ Chiffres 0...9

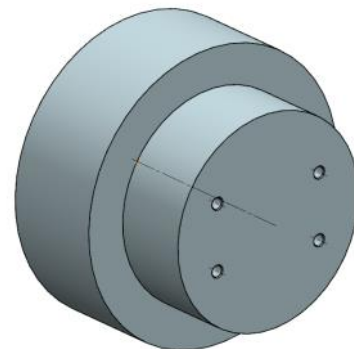
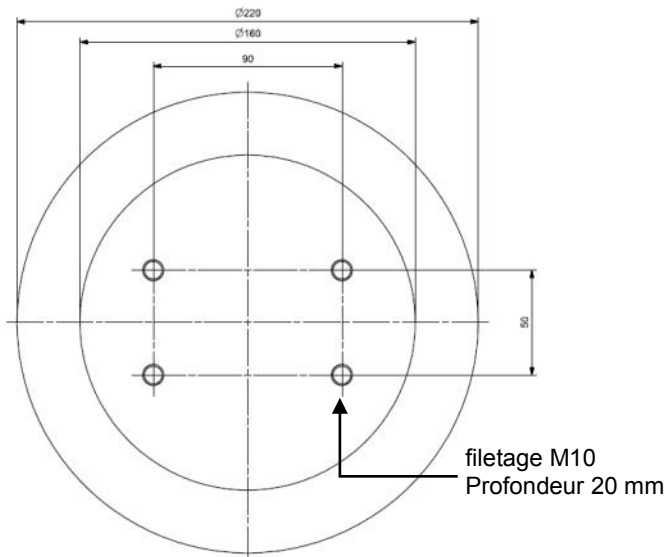
⇒ Tirets bas _

Les mots clés réservés, par exemple "RET", ne peuvent pas être utilisés dans les noms.

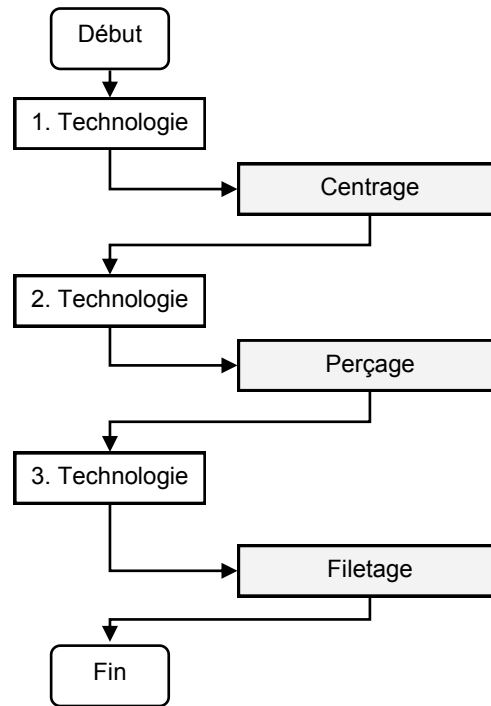
Tâche :

Le programme de base existant du module M200 est divisé en sous-programmes, en fonction de l'usinage. Le programme principal et les sous-programmes se terminent par l'extension MPF et doivent se trouver dans le même répertoire.

Le programme de base est désormais optimisé à l'aide de sous-programmes.

Schéma :

Remarques

Organigramme :Programme échantillon :**Programme principal**

```

N10 G17 G60 G90 G94
N20 G54
N30 WORKPIECE(,,,"CYLINDER",192,0,-120,-100,160)
N40 G18 ;TURNINGPLANE
N50 SPOS=0 ;SPINDLEPOSITIONING
N60 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N70 TRANSMIT ;TRANSFORM MILLING INTO TURNING
N80 DIAMOF ;PROGRAMMING IN RADIUS
N90 G17 ;MILLINGPLANE
N100 Z400 ZS2=700 F100
;SUBPROGRAM CALL
N110 _CENTERING
N120 T="CENTERDRILL"
N130 S3=1000 M3=3
;SUBPROGRAM CALL
N140 _DRILLING
N150 T="DRILL_8_5"
N160 S3=1500 M3=3
;SUBPROGRAM CALL
N180 T="TAP_10"
N190 S200 M3=3
N200 G0 Z100
N210 M30
  
```

1. Technologie

2. Technologie

3. Technologie

Remarques

Sous-programme de centrage

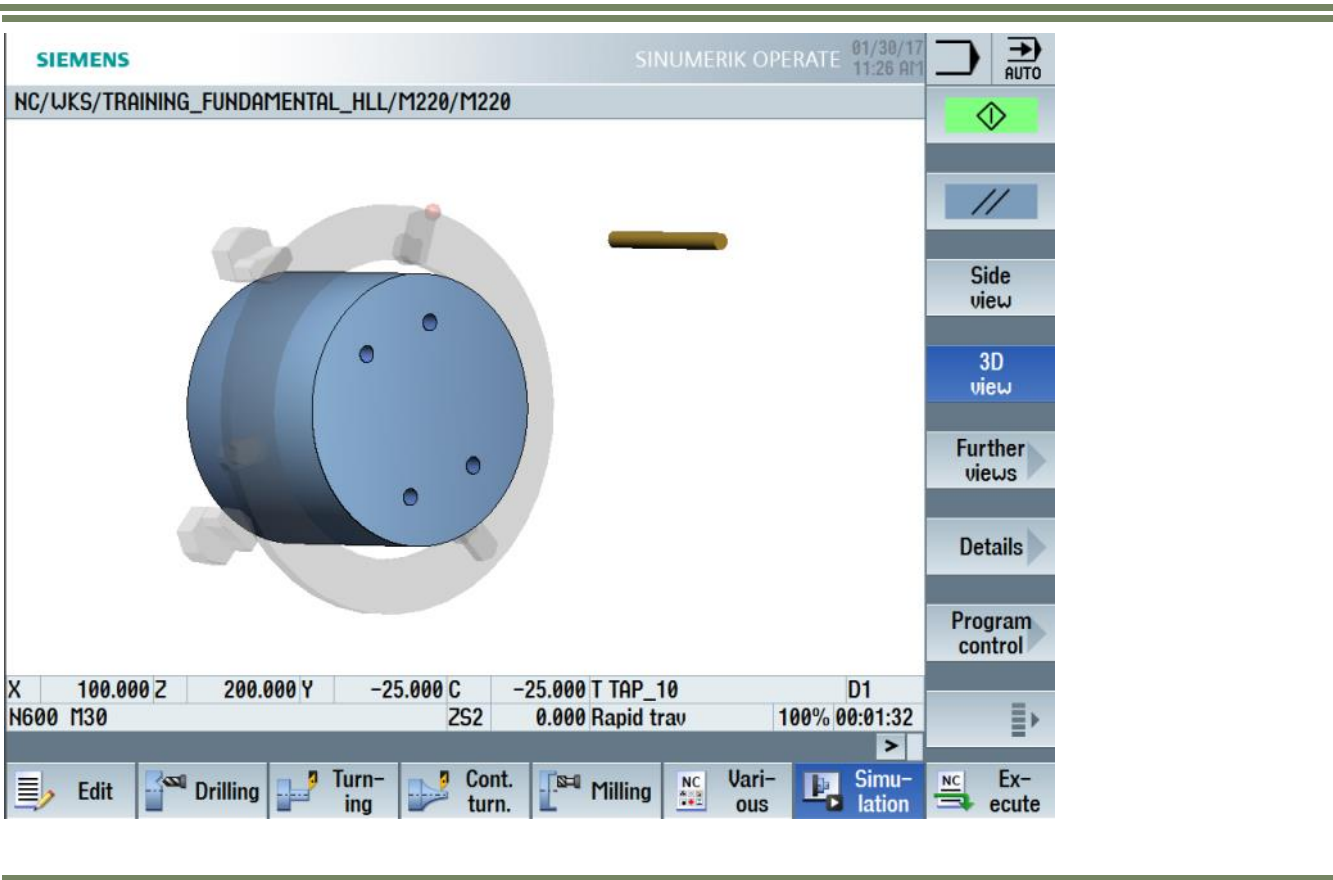
N10 G0 Z3
 N20 G0 X-45 Y25
 N30 G1 Z-2 F140
 N40 G0 Z3
 N50 G0 X45 Y25
 N60 G1 z-2
 N70 G0 Z3
 N80 G0 X45 Y-25
 N90 G1 z-2
 N100 G0 Z3
 N110 G0 X-45 Y-25
 N120 G1 z-2
 N130 G0 Z3
 N140 G0 X100 Z200
 N150 M17

Sous-programme de perçage

N10 G0 Z3
 N20 G0 X-45 Y25
 N30 G1 Z-22 F150
 N40 G0 Z3
 N50 G0 X45 Y25
 N60 G1 z-22
 N70 G0 Z3
 N80 G0 X45 Y-25
 N90 G1 z-22
 N100 G0 Z3
 N110 G0 X-45 Y-25
 N120 G1 z-22
 N130 G0 Z3
 N140 G0 X100 Z200
 N150 RET

Sous-programme de filetage

N10 G0 Z3
 N20 G0 X-45 Y25
 N30 G331 Z-20 K1.5 S200
 N40 G332 Z3 K1.5 S400
 N50 G0 X45 Y25
 N60 G331 Z-20 K1.5 S200
 N70 G332 Z3 K1.5 S400
 N80 G0 X45 Y-25
 N90 G331 Z-20 K1.5 S200
 N100 G332 Z3 K1.5 S400
 N110 G0 X-45 Y-25
 N120 G331 Z-20 K1.5 S200
 N130 G332 Z3 K1.5 S400
 N140 G0 X100 Z200
 N150 M17



Remarques

Remarques

Description du module :

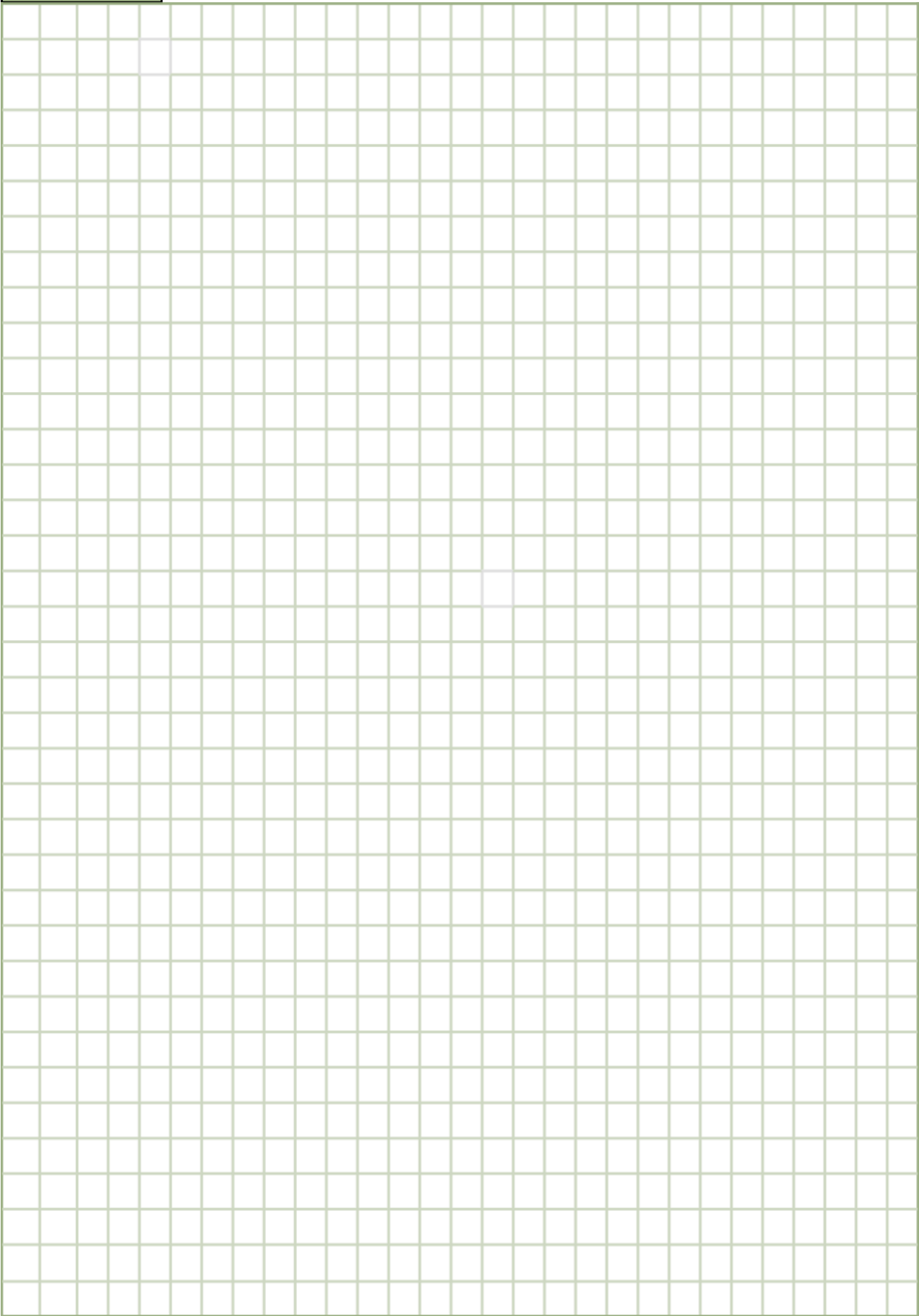
Ce module décrit le travail avec des variables R (paramètres arithmétiques).

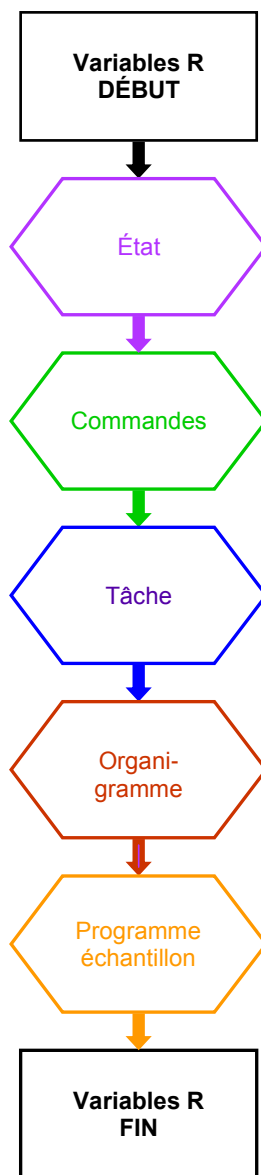
Cas d'utilisation :

Après avoir complété le module, vous pourrez configurer plus efficacement des programmes CNC, programmer des requêtes STATUS et ACTUAL et les visualiser. Les programmes CNC deviendront plus simples et plus transparents.

Explication d'un travail avec des variables R

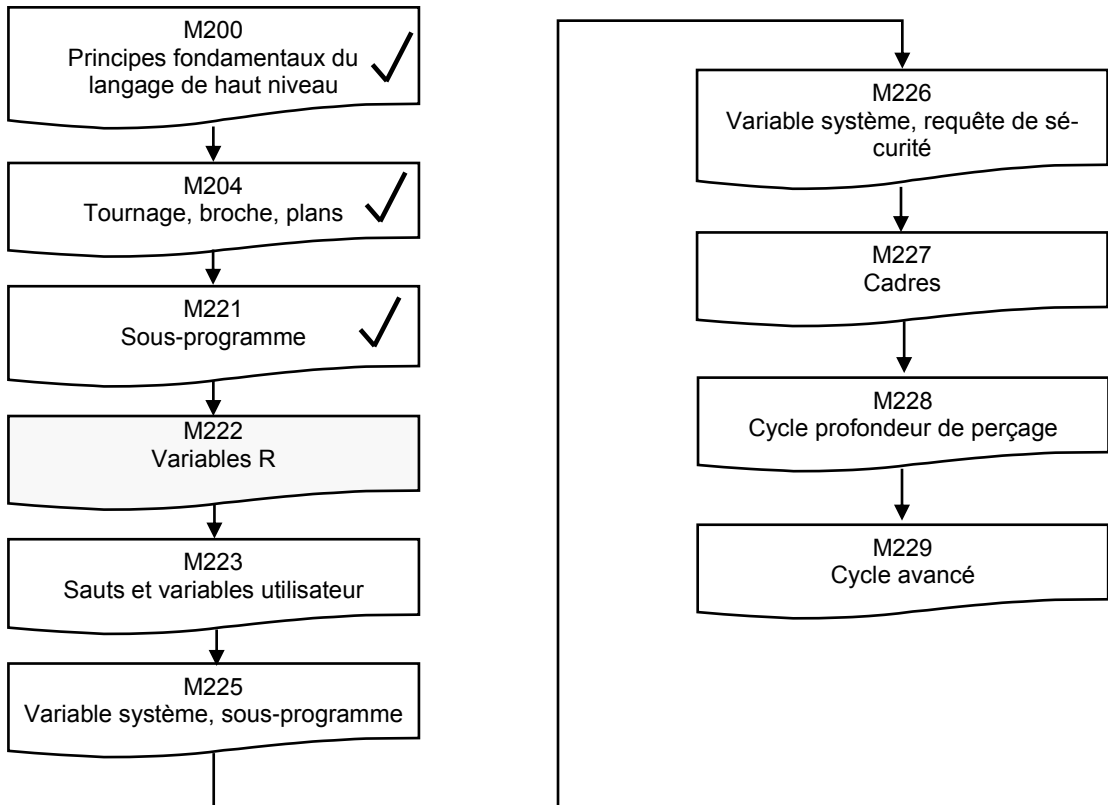
Insertion des contenus basée sur un exemple





Remarques

État :



Commandes :

Commande	Signification
R...	Paramètres arithmétiques prédéfinis pour lesquels la lettre "R" est suivie d'un numéro de classification.

Les paramètres arithmétiques sont disponibles pour chaque contrôle SINUMERIK et peuvent être utilisés par les utilisateurs/programmeurs.

Le nombre de paramètres arithmétiques disponibles varie et peut être défini par une donnée machine. Normalement, 99 variables sont disponibles.

Veuillez vérifier les spécifications du fabricant de la machine.

Fonction :

Les paramètres arithmétiques peuvent être utilisés lorsqu'un programme NC doit être configuré de manière flexible, car les valeurs varient à l'intérieur du programme en fonction de la pièce.

Il est possible d'attribuer à ces paramètres des adresses NC arbitraires dans un programme, avec une valeur flexible. Les valeurs attribuées doivent être numériques.

Il est important que des valeurs correctes soient attribuées à ces variables, car leur validité ne sera pas vérifiée en standard.

Les variables R sont des variables qui sont conservées (mémorisées) une fois que la machine est éteinte.

Les variables R sont locales, visibles uniquement sur le canal actuel.

Remarques

Conditions requises d'une variable R :

⇒ La valeur d'une variable R doit être attribuée dans un bloc

Exemple :

R12=100

⇒ Un maximum de huit chiffres après la virgule sont autorisés.

⇒ Le point décimal peut être ignoré avec les nombres entiers.

⇒ Le signe peut être ignoré avec les nombres positifs.

⇒ Les fonctions arithmétiques peuvent être exécutées avec des variables R - la notation mathématique s'applique.

⇒ Une attribution peut être effectuée sur pratiquement toutes les adresses avec des variables R, à l'exception de N, G et L.

Variables R dans SINUMERIK :**Affichage des variables R sur le contrôle**

SIEMENS		SINUMERIK OPERATE		01/31/17 1:22 PM		AUTO
R variables						
R 0	0	R 20	0	R 40	0	
R 1	0	R 21	0	R 41	0	
R 2	0	R 22	0	R 42	0	R variables
R 3	0	R 23	0	R 43	0	
R 4	0	R 24	0	R 44	0	
R 5	0	R 25	0	R 45	0	Global GUD
R 6	0	R 26	0	R 46	0	
R 7	0	R 27	0	R 47	0	Channel GUD
R 8	0	R 28	0	R 48	0	
R 9	0	R 29	0	R 49	0	
R 10	0	R 30	0	R 50	0	Local LJD
R 11	0	R 31	0	R 51	0	
R 12	0	R 32	0	R 52	0	
R 13	0	R 33	0	R 53	0	
R 14	0	R 34	0	R 54	0	
R 15	0	R 35	0	R 55	0	
R 16	0	R 36	0	R 56	0	
R 17	0	R 37	0	R 57	0	Search
R 18	0	R 38	0	R 58	0	
R 19	0	R 39	0	R 59	0	

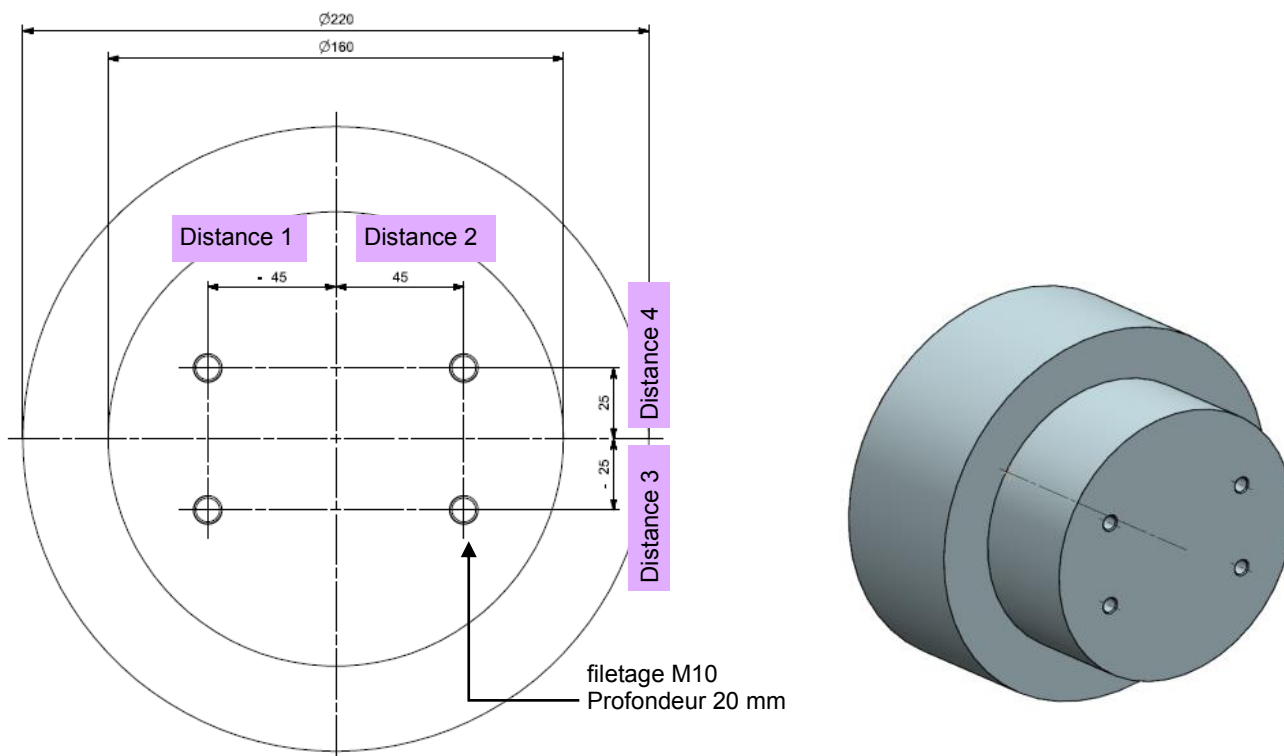
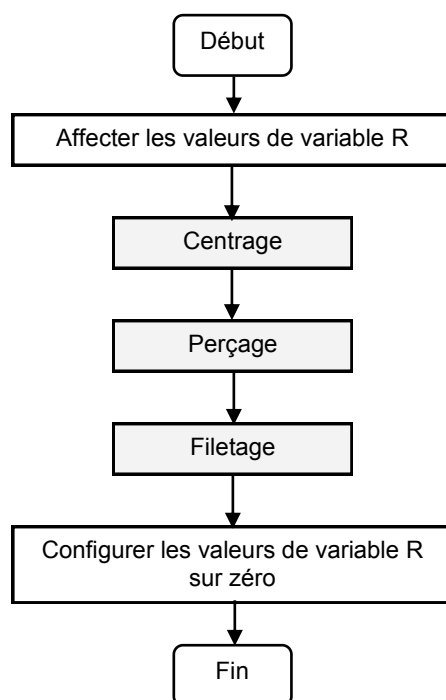
Tool list | Tool wear | Magazine | Work offset | **User variable** | SD Setting data

Tâche :

1. Tâche :

Les distances de trou par rapport au zéro de la pièce doivent être programmées de façon variable dans le programme.

Remarques

Schéma :Organigramme :

Un programme peut être utilisé pour différentes pièces avec une programmation de variables R.

Remarques

Programme échantillon

```

;PLEASE ASSIGN CORRECT VALUES TO THE R VARIABLES
;R1= DISTANCE 1 IN THE DRAWING
;R2= DISTANCE 2 IN THE DRAWING
;R3= DISTANCE 3 IN THE DRAWING
;R4= DISTANCE 4 IN THE DRAWING
N10 R1=-45
N20 R2=45
N30 R3=-25
N40 R4=25
N50 MSG("FILLED IN THE CORRECT VALUES?")
N60 M0
N70 MSG()
N80 G17 G60 G90 G94
N90 G54
N100 WORKPIECE(,,,"CYLINDER",192,0,-120,-100,160)
N110 G18 ;TURNINGPLANE
N120 SPOS=0 ;SPINDLEPOSITIONING
N130 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N140 T="CENTERDRILL"
N150 S3=1000 M3=3
N160 TRANSMIT ;TRANSFORM MILLING INTO TURNING
N170 DIAMOF ;PROGRAMMING IN RADIUS
N180 G17 ;MILLINGPLANE
N190 F140
N200 G0 Z3
N210 G0 X=R1 Y=R4
N220 G1 Z-2
N230 G0 Z3

```

Affecter les valeurs de variable R

Centrage

```

N240 G0 X=R2 Y=R4
N250 G1 z-2
N260 G0 Z3
N270 G0 X=R2 Y=R3
N280 G1 z-2
N290 G0 Z3
N300 G0 X=R1 Y=R3
N310 G1 z-2
N320 G0 Z3
N330 G0 X100 Z200
N340 T="DRILL_8_5"
N350 S3=1500 M3=3
N360 F180
N370 G0 Z3
N380 G0 X=R1 Y=R4
N390 G1 Z-22
N400 G0 Z3
N410 G0 X=R2 Y=R4
N420 G1 z-22
N430 G0 Z3
N440 G0 X=R2 Y=R3
N450 G1 z-22
N460 G0 Z3
N470 G0 X=R1 Y=R3
N480 G1 z-22
N490 G0 Z3
N500 G0 X100 Z200
N510 T="TAP_10" ;TAPPING TOOL
N520 S200 M2=3
N530 G0 Z3

```

Perçage

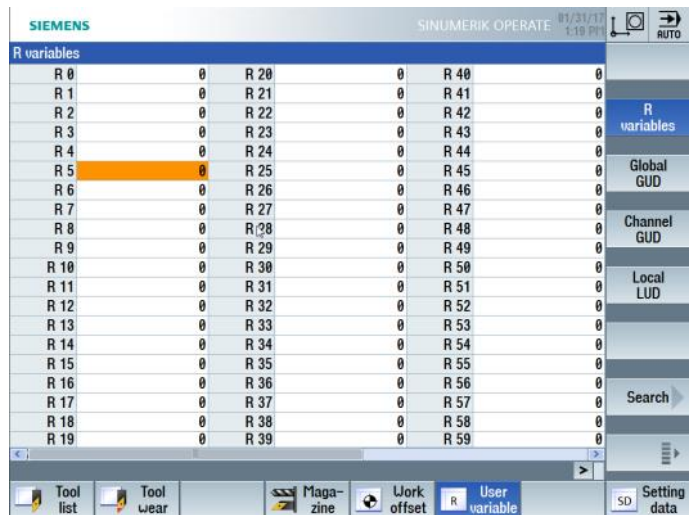
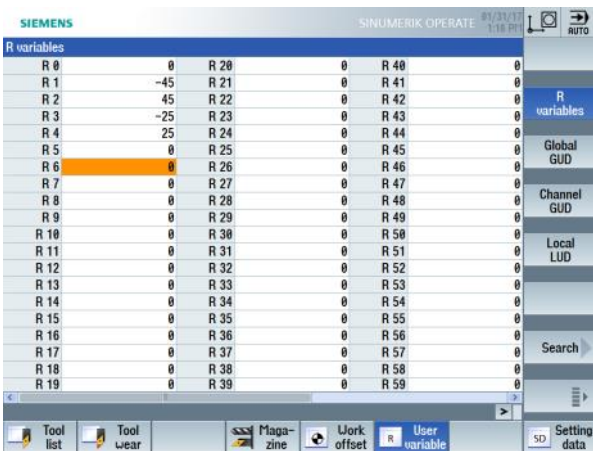
Filetage

Remarques

N530 G0 Z3
 N540 G0 X=R1 Y=R4
 N550 G331 Z-20 K1.5 S200
 N560 G332 Z3 K1.5 S400
 N570 G0 X=R2 Y=R4
 N580 G331 Z-20 K1.5 S200
 N590 G332 Z3 K1.5 S400
 N600 G0 X=R2 Y=R3
 N610 G331 Z-20 K1.5 S200
 N620 G332 Z3 K1.5 S400
 N630 G0 X=R1 Y=R3
 N640 G331 Z-20 K1.5 S200
 N650 G332 Z3 K1.5 S400
 N660 G0 X100 Z200
 N670 R1=0
 N680 R2=0
 N690 R3=0
 N700 R4=0
 N710 M30

Réinitialiser les variables R

Les valeurs saisies sont attribuées aux variables R dans la liste des variables R, puis désélectionnées à la fin du programme.



Remarques

Textes de message :

Commande	Signification
MSG	Des textes arbitraires, ainsi que des valeurs de variable, peuvent être envoyés sur la ligne d'alarme dans la zone d'opération Automatic .

Des messages peuvent être programmés par l'opérateur avec la commande MSG, pour s'afficher pendant l'exécution du programme.

Des textes arbitraires et des valeurs de variable peuvent être également affichés via la commande MSG.

Exemple de texte de message :

MSG("Ceci est un texte d'exemple")

Ce texte reste actif et visible jusqu'à la fin du programme.

MSG()

Cette commande désélectionne le texte actuellement affiché à une position arbitraire dans le programme.

Exemple de texte de message avec une variable :

MSG("Valeur de la variable R1="<<R1)

Le texte entre guillemets peut être arbitraire "Valeur de la variable R1=". Deux symboles "inférieur à" << affichent la valeur dans le cas de la variable R R1.

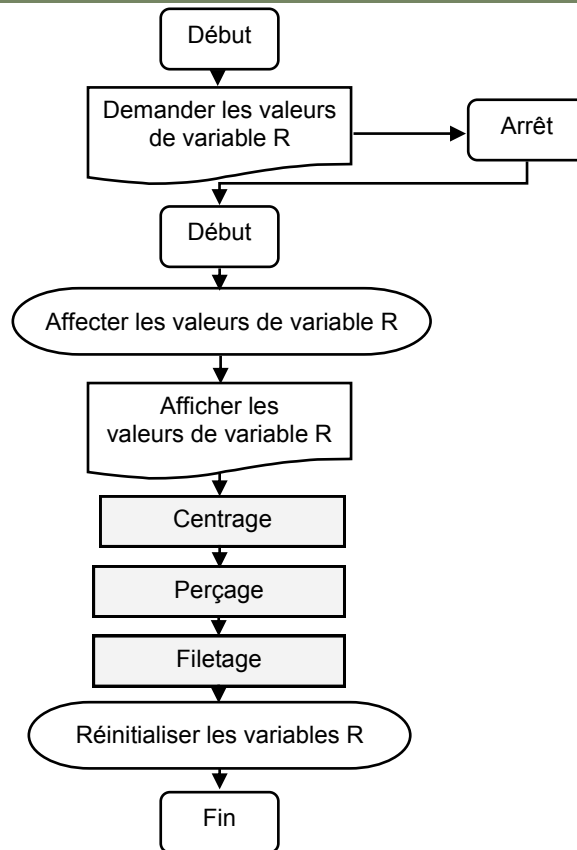
Description de la tâche

2 Tâche :

Un texte de message s'affiche pour l'opérateur si les valeurs ont été saisies correctement dans les variables R.

À la fin, les valeurs actuelles des variables R s'affichent pour l'opérateur dans la zone d'opération Automatic.

Remarques

Organigramme :

```

; PLEASE ASSIGN CORRECT VALUES TO THE R VARIABLES
; R1= DISTANCE 1 IN THE DRAWING
; R2= DISTANCE 2 IN THE DRAWING
; R3= DISTANCE 3 IN THE DRAWING
; R4= DISTANCE 4 IN THE DRAWING
N10 R1=-45
N20 R2=45
N30 R3=-25
N40 R4=25
MSG("FILLED IN THE CORRECT VALUES?")
M0
MSG()
MSG("R1="<<<R1<<"__R2="<<<R2<<"__R3="<<<R3<<"__R4="<<<R4)
N10 G17 G60 G90 G94
.
.
.
...
M30; DELETES MESSAGE TEXT
  
```

Afficher les
valeurs de variable R

Arrêt

Afficher les
valeurs de variable R

Remarques

1er texte de message

The screenshot shows the CNC control interface with the following data:

Work	Position [mm]	Dist-to-go	T,F,S
X	206.155	0.000	T TAP_10 ∅ 10.000
Y	-0.007	0.000	7 D1 Z 100.000
Z	200.000	0.000	F 0.000 X 0.000
ZS2	0.000	0.000	0.000 mm/min 85%
C	345.965°	0.000	S1 0

Additional information: MRD Stop: M0/M1 active, MSG("FILLED IN THE CORRECT VALUES?")

Ce texte est supprimé en appuyant sur Cycle Start.

2e texte de message

The screenshot shows the CNC control interface with the following data:

Work	Position [mm]	Dist-to-go	T,F,S
X	45.000	0.000	T CENTERDRILL ∅ 12.000
Y	-25.000	0.000	6 D1 Z 100.000
Z	0.429	-2.305	F 23.958 X 0.000
Y1	-0.007	0.000	140.000 mm/min 85%
ZS2	0.000	0.000	S2 1000
			Master 1000 100%

Additional information: R1=-45_R2=45_R3=-25_R4=25, MSG("R1=-45_R2=45_R3=-25_R4=25")

Ce texte reste visible jusqu'à la fin du programme M30.

Remarques

Remarques

Description du module :

Dans ce module, vous apprendrez à définir vos propres variables et à travailler avec des instructions de saut.

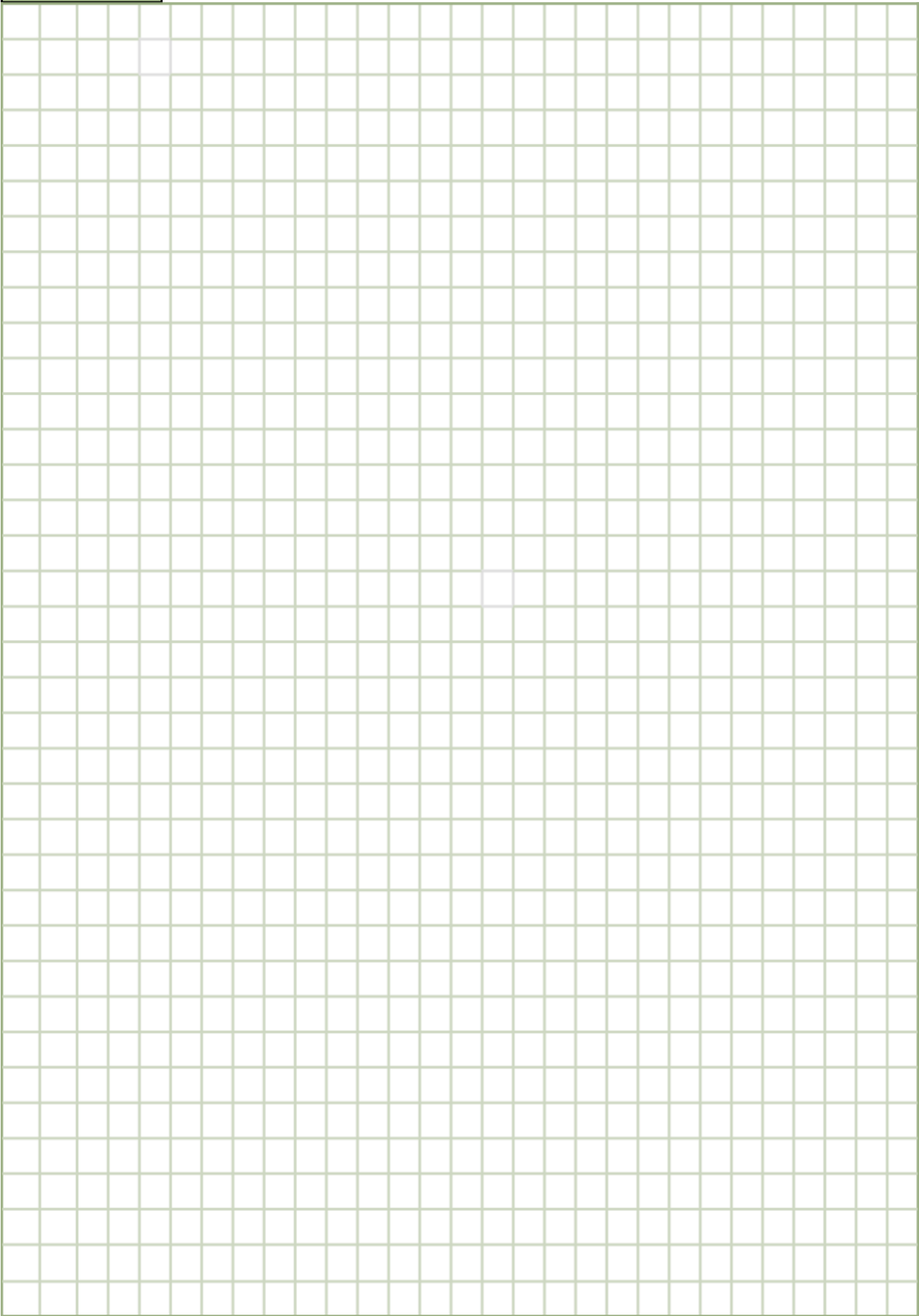
Objectif du module :

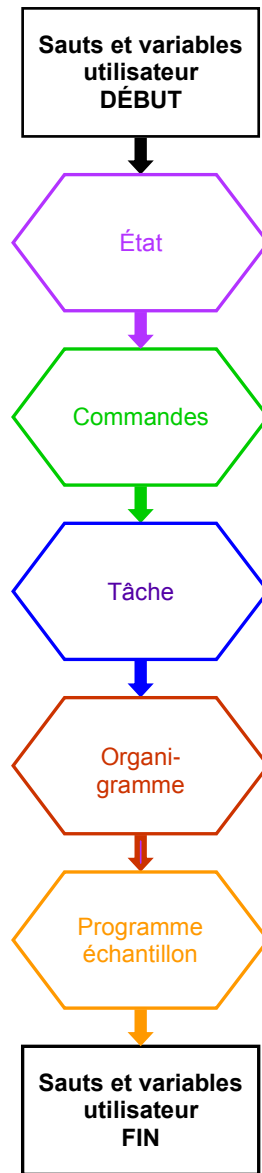
Après avoir complété ce module, vous pourrez définir des variables utilisateur simples et programmer des requêtes avec des instructions de saut. Vous travaillerez de manière plus efficace et plus flexible.

Explication des variables

Explication des sauts

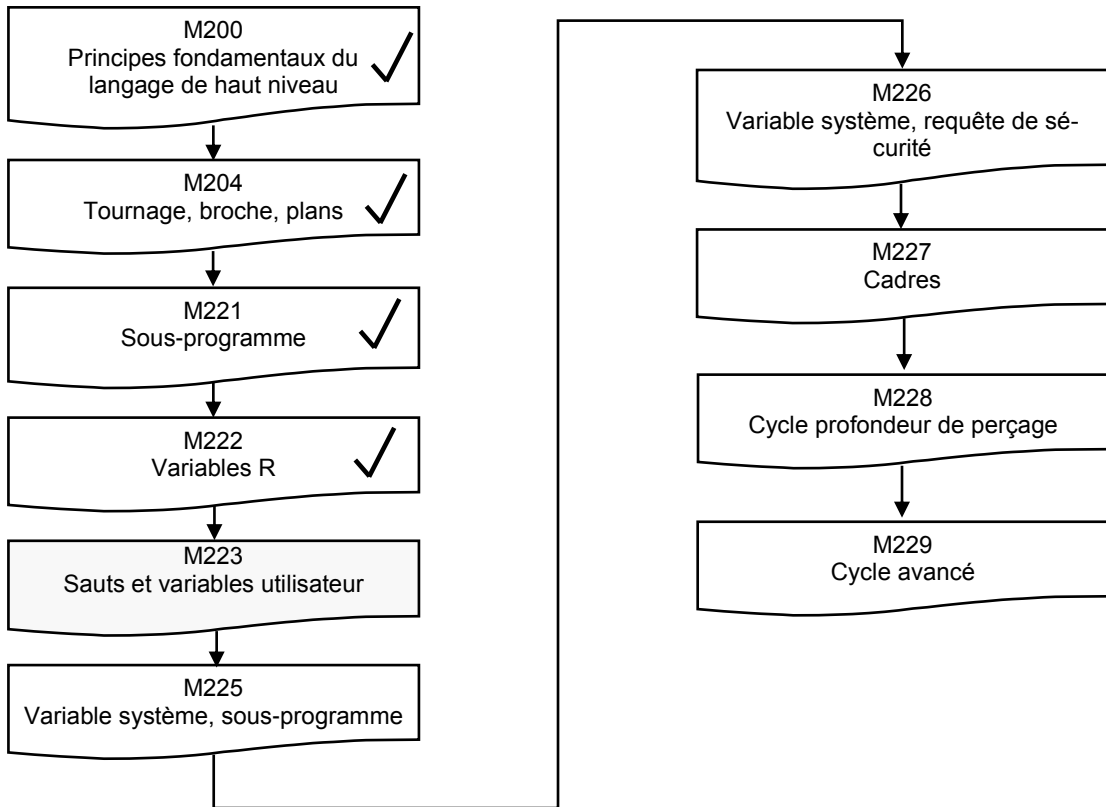
Programmation des requêtes





Remarques

État :



Commandes :

Commande	Signification
DEF INT	Variables définies par l'utilisateur
IF	Conditions
LABEL	Étiquettes de saut
GOTO; GOTOF; GOTOB	Instructions de saut

DEF INT :

Vous pouvez définir vos propres variables et leur attribuer des valeurs avec cette commande. Les variables dites utilisateur.

- ⇒ Les variables utilisateur doivent toujours commencer par un tiret bas, afin de s'assurer qu'aucun nom préattribué n'est utilisé.
- ⇒ Une variable utilisateur ne peut pas dépasser 15 caractères.
- ⇒ La notation est toujours la même (tous les caractères en majuscules).
- ⇒ INT identifie toujours les nombres entiers. Il existe différents types de variable, qui sont décrits en détail dans le manuel COMPLETE HIGH LEVEL LANGUAGE.
- ⇒ Une valeur est attribuée à la variable à l'aide du signe égal (VALUE =).
- ⇒ La requête sur la signification d'une variable s'effectue à l'aide de deux signes égal (IF VALUE == 1).

IF :

Cette commande est un mot clé auquel est toujours rattachée une condition. De cette façon, le programmeur peut modifier la séquence des programmes NC afin d'intégrer une structure de contrôle.

Remarques

Étiquette :

Les étiquettes de saut peuvent être définies avec une étiquette déclarée, qui est référencée depuis d'autres positions dans le même programme. Le nom de l'étiquette peut être sélectionné de manière arbitraire. Il est pratique d'utiliser un nom d'étiquette "parlant", par exemple FILETAGE.

Si une étiquette est référencée comme destination du saut, un caractère deux points doit être placé derrière le mot.

ETIQUETTE_1:

Cette convention désigne l'étiquette en tant que telle.

GOTO/GOTOF/GOTOB :

GOTO est une commande de saut directe qui est toujours utilisée en conjonction avec une étiquette de saut.

La direction du saut est définie par **GOTOF** ou **GOTOB**. Cela raccourcit le temps de recherche.

Avec le même nom, la destination du saut peut être une étiquette dans le programme ou le sous-programme actif, dont l'extension se termine par MPF ou SPF.

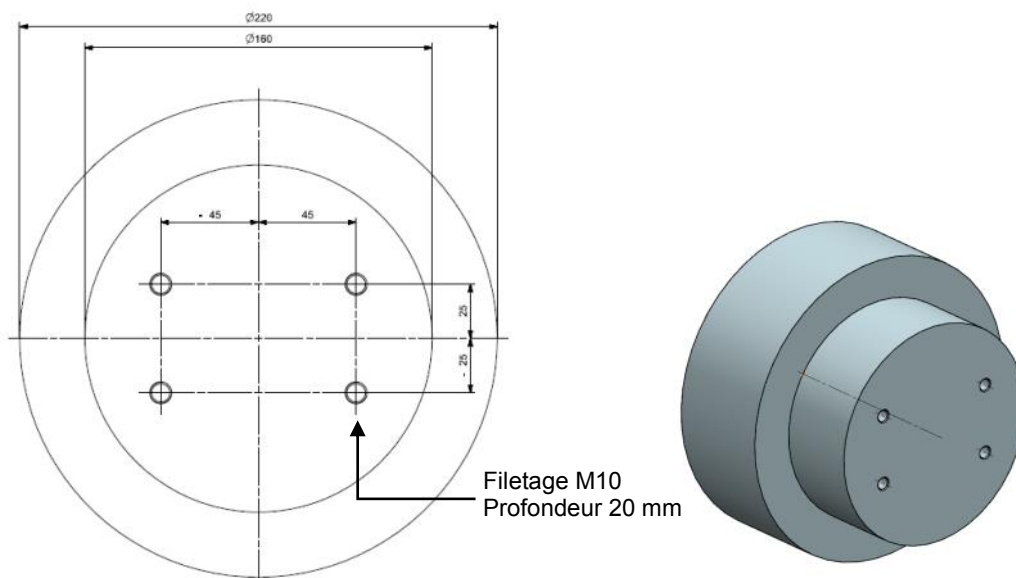
Des noms uniques doivent être utilisés lorsque vous travaillez avec des identifiants "parlants".

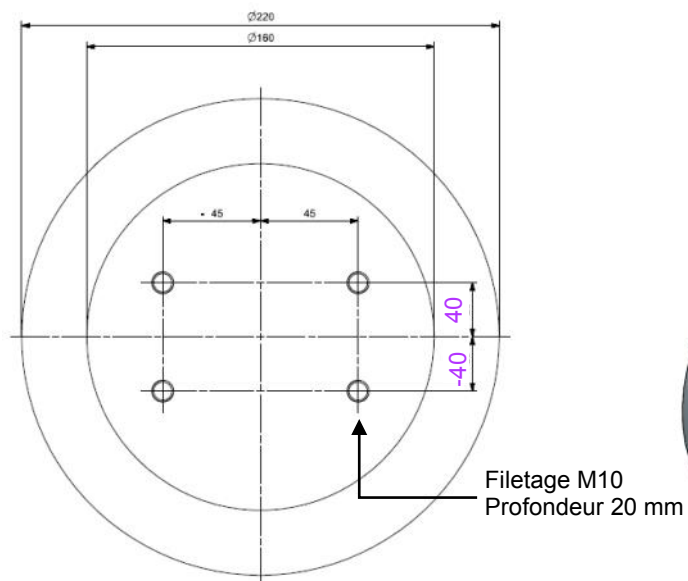
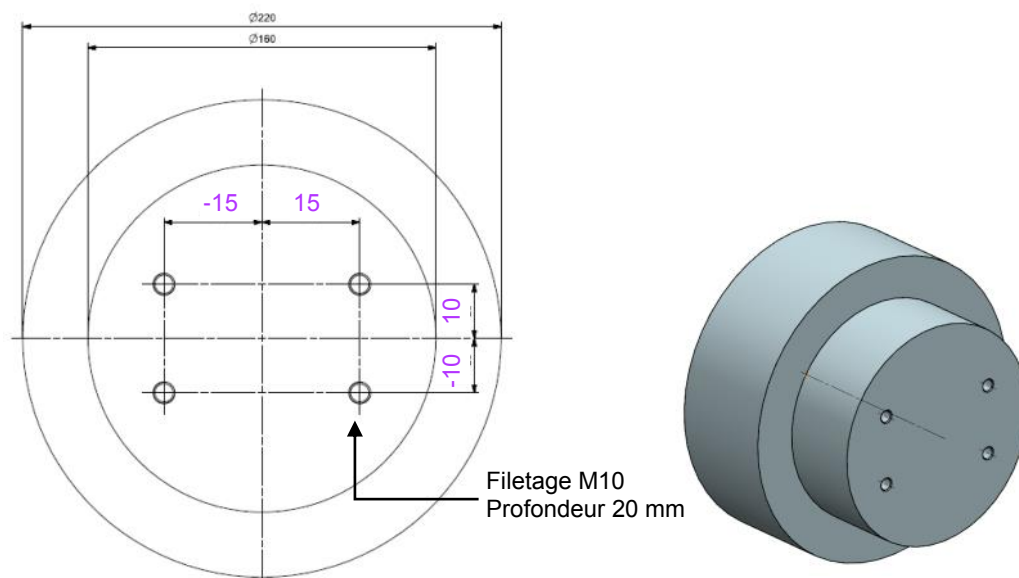
Si la destination du saut n'est pas trouvée, une alarme 14080 "Jump destination not found" est émise.

Tâche :**Tâche :**

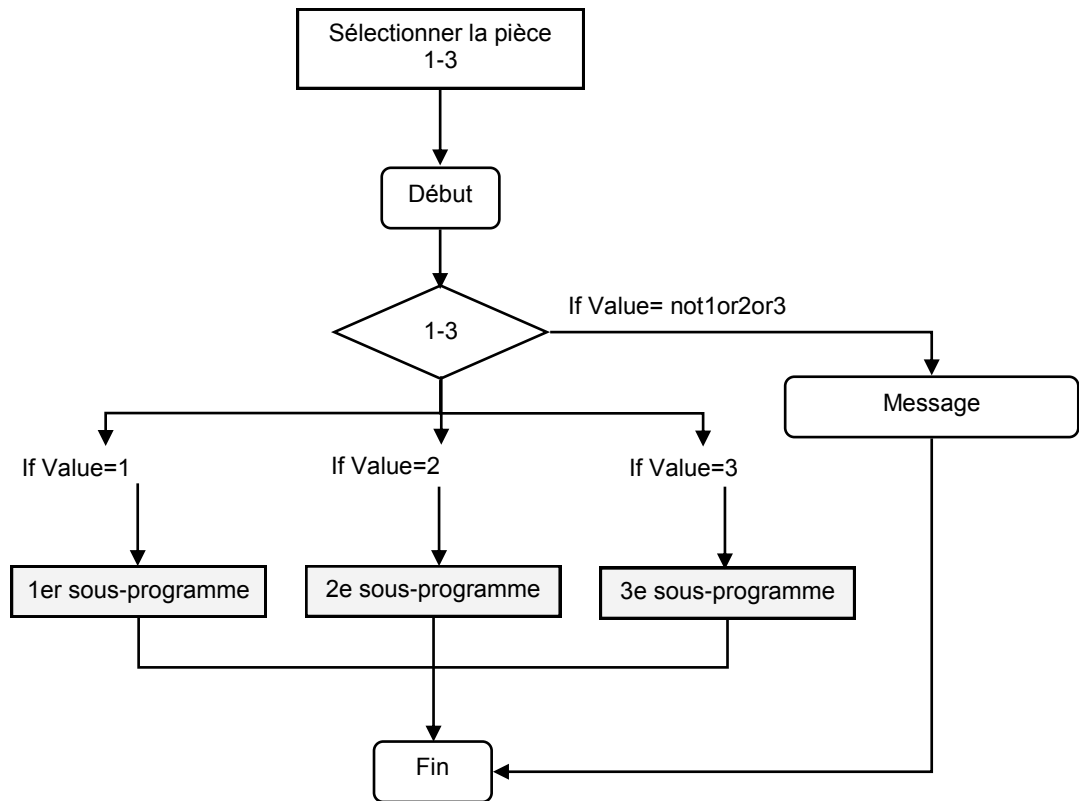
Le programme de base avec ses sous-programmes du module 201 est utilisé

- ⇒ L'opérateur pourra choisir entre trois programmes avec différentes distances de trou.
- ⇒ Le sous-programme correspondant, avec les distances de perçage appropriées, doit être sélectionné en saisissant une valeur comprise entre 1 et 3.
- ⇒ Si aucune valeur entre 1 et 3 n'est sélectionnée, le programme est abandonné. Un message s'affiche à propos de la saisie incorrecte.

1er sous-programme**Remarques**

2e programme**3e programme**

Remarques

Organigramme :Programme échantillon :**Programme principal :**

```

N10 DEF INT _WORKPIECE
;ENTER NUMBER OF THE WORKPIECE IN THE NEXT BLOCK
N20 WORKPIECE=3
;QUERY VALUE WITH ASSIGNMENT JUMP
N30 IF _WORKPIECE==1 GOTOF LABEL_1
N40 IF _WORKPIECE==2 GOTOF LABEL_2
N50 IF _WORKPIECE==3 GOTOF LABEL_3
N60 MSG("INCORRECT VALUE") ;DISPLAY IF VALUE NOT 1-3
N70 M0
N80 M30 ;END OF PROGRAM IF VALUE INCORRECT
N90 LABEL_1: ;JUMP LABEL
N100 PROGRAM_1
N110 M30
N120 LABEL_2:
N130 PROGRAM_2
N140 M30
N150 LABEL_3:
N160 PROGRAM_3
N170 M30
  
```

Sélectionner la pièce
1-3

If Value=1

If Value=2

If Value=3

Remarques

Sous-programme pièce 1

```
N10 G17 G60 G90 G94
N20 G54
N30 WORKPIECE(,,, "CYLINDER",192,0,-120,-100,160)
N40 G18 ;TURNINGPLANE
N50 SPOS=0 ;SPINDLEPOSITIONING
N60 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N70 T="CENTERDRILL"
N80 S3=1000 M3=3
N90 TRANSMIT ;TRANSFORM MILLING INTO TURNING
N100 DIAMOF ;PROGRAMMING IN RADIUS
N110 G17 ;MILLINGPLANE
N120 F140
N130 G0 Z3
N140 G0 X-45 Y25
N150 G1 Z-2
N160 G0 Z3
N170 G0 X45 Y25
N180 G1 z-2
N190 G0 Z3
N200 G0 X45 Y-25
N210 G1 z-2
N220 G0 Z3
N230 G0 X-45 Y-25
N240 G1 z-2
N250 G0 Z3
N260 G0 X100 Z200
N270 T="DRILL_8_5"
N280 S3=1500 M3=3
N290 F180
```

```
N300 G0 Z3
N310 G0 X-45 Y25
N320 G1 Z-22
N330 G0 Z3
N340 G0 X45 Y25
N350 G1 z-22
N360 G0 Z3
N370 G0 X45 Y-25
N380 G1 z-22
N390 G0 Z3
N400 G0 X-45 Y-25
N410 G1 z-22
N420 G0 Z3
N430 G0 X100 Z200
N440 T="TAP_10"
N450 S200 M3=3
N460 G0 Z3
N470 G0 X-45 Y25
N480 G331 Z-20 K1.5 S200
N490 G332 Z3 K1.5 S400
N500 G0 X45 Y25
N510 G331 Z-20 K1.5 S200
N520 G332 Z3 K1.5 S400
N530 G0 X45 Y-25
N540 G331 Z-20 K1.5 S200
N550 G332 Z3 K1.5 S400
N560 G0 X-45 Y-25
N570 G331 Z-20 K1.5 S200
N580 G332 Z3 K1.5 S400
N590 G0 X100 Z200
N600 M17
```

Remarques

Sous-programme pièce 2

```
N10 G17 G60 G90 G94
N20 G54
N30 WORKPIECE(,,, "CYLINDER",192,0,-120,-100,160)
N40 G18 ;TURNINGPLANE
N50 SPOS=0 ;SPINDLEPOSITIONING
N60 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N70 T="CENTERDRILL"
N80 S3=1000 M3=3
N90 TRANSMIT ;TRANSFORM MILLING INTO TURNING
N100 DIAMOF ;PROGRAMMING IN RADIUS
N110 G17 ;MILLINGPLANE
N120 F140
N130 G0 Z3
N140 G0 X-45 Y40
N150 G1 Z-2
N160 G0 Z3
N170 G0 X45 Y40
N180 G1 z-2
N190 G0 Z3
N200 G0 X45 Y-40
N210 G1 z-2
N220 G0 Z3
N230 G0 X-45 Y-40
N240 G1 z-2
N250 G0 Z3
N260 G0 X100 Z200
N270 T="DRILL_8_5"
N280 S3=1500 M3=3
N290 F180
N300 G0 Z3
```

```
N310 G0 X-45 Y40
N320 G1 Z-22
N330 G0 Z3
N340 G0 X45 Y40
N350 G1 z-22
N360 G0 Z3
N370 G0 X45 Y-40
N380 G1 z-22
N390 G0 Z3
N400 G0 X-45 Y-40
N410 G1 z-22
N420 G0 Z3
N430 G0 X100 Z200
N440 T="TAP_10"
N450 S200 M3=3
N460 G0 Z3
N470 G0 X-45 Y40
N480 G331 Z-20 K1.5 S200
N490 G332 Z3 K1.5 S400
N500 G0 X45 Y40
N510 G331 Z-20 K1.5 S200
N520 G332 Z3 K1.5 S400
N520 G332 Z3 K1.5 S400
N530 G0 X45 Y-40
N540 G331 Z-20 K1.5 S200
N550 G332 Z3 K1.5 S400
N560 G0 X-45 Y-40
N570 G331 Z-20 K1.5 S200
N580 G332 Z3 K1.5 S400
N590 G0 X100 Z200
N600 M17
```

Remarques

Sous-programme pièce 3

```

N10 G17 G60 G90 G94
N20 G54
N30 WORKPIECE(,,, "CYLINDER",192,0,-120,-100,160)
N40 G18 ;TURNINGPLANE
N50 SPOS=0 ;SPINDLEPOSITIONING
N60 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N70 T="CENTERDRILL"
N80 S3=1000 M3=3
N90 TRANSMIT ;TRANSFORM MILLING INTO TURNING
N100 DIAMOF ;PROGRAMMING IN RADIUS
N110 G17 ;MILLINGPLANE
N120 F140
N130 G0 Z3
N140 G0 X-15 Y10
N150 G1 Z-2
N160 G0 Z3
N170 G0 X15 Y10
N180 G1 z-2
N190 G0 Z3
N200 G0 X15 Y-10
N210 G1 z-2
N220 G0 Z3
N230 G0 X-15 Y-10
N240 G1 z-2
N250 G0 Z3
N260 G0 X100 Z200
N270 T="DRILL_8_5"
N280 S3=1500 M3=3
N290 F180

```

```

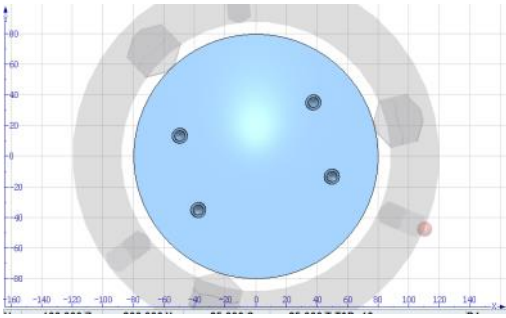
N300 G0 Z3
N310 G0 X-15 Y10
N320 G1 Z-22
N330 G0 Z3
N340 G0 X15 Y10
N350 G1 z-22
N360 G0 Z3
N370 G0 X15 Y-10
N380 G1 z-22
N390 G0 Z3
N400 G0 X-15 Y-10
N410 G1 z-22
N420 G0 Z3
N430 G0 X100 Z200
N440 T="TAP_10"
N450 S200 M3=3
N460 G0 Z3
N470 G0 X-15 Y10
N480 G331 Z-20 K1.5 S200
N490 G332 Z3 K1.5 S400
N500 G0 X15 Y10
N510 G331 Z-20 K1.5 S200
N520 G332 Z3 K1.5 S400
N530 G0 X15 Y-10
N540 G331 Z-20 K1.5 S200
N550 G332 Z3 K1.5 S400
N560 G0 X-15 Y-10
N570 G331 Z-20 K1.5 S200
N580 G332 Z3 K1.5 S400
N590 G0 X100 Z200
N600 M17

```

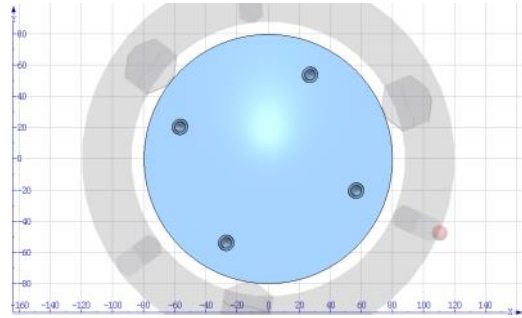
Remarques

Simulation :

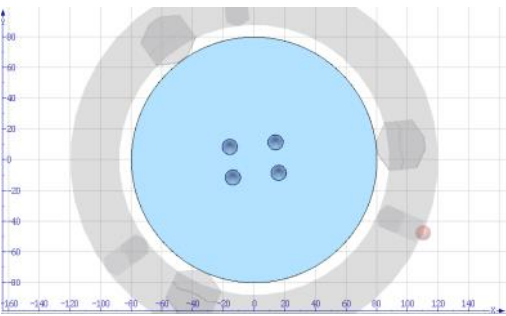
Pièce 1



Pièce 2



Pièce 3

**Tâche additionnelle :**

Le programme doit être optimisé. Pour les trois options, le sous-programme 1 est utilisé.

⇒ Astuce : Solution : Utiliser le paramètre R.

Remarques

Remarques

Description du module :

Ce module décrit les bases des variables système et la façon de travailler avec elles dans un programme CNC.

Objectif du module :

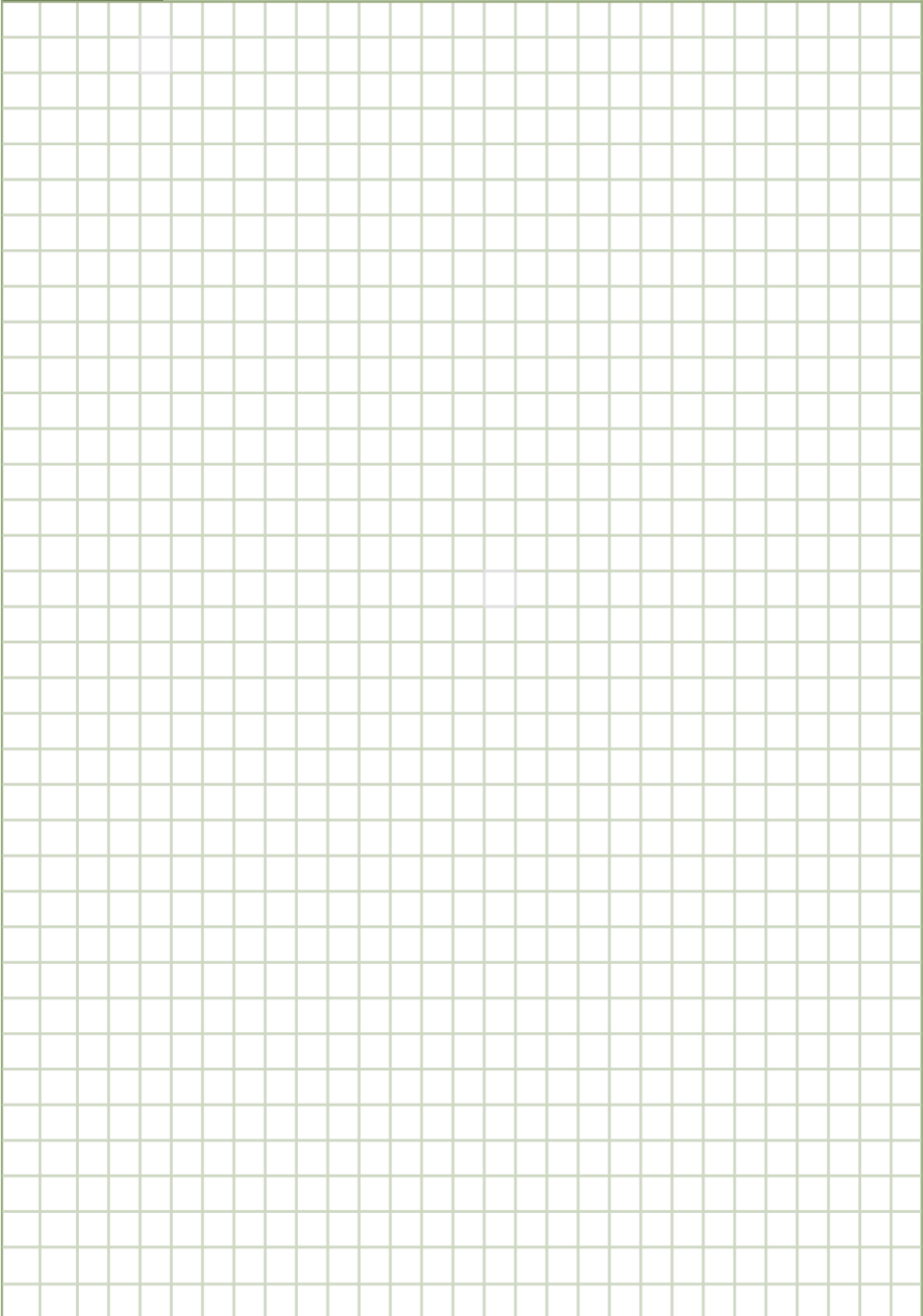
Après avoir complété ce module, vous pourrez programmer un simple cycle à l'aide d'une variable système. Pour chaque tâche, vous pouvez élaborer une solution

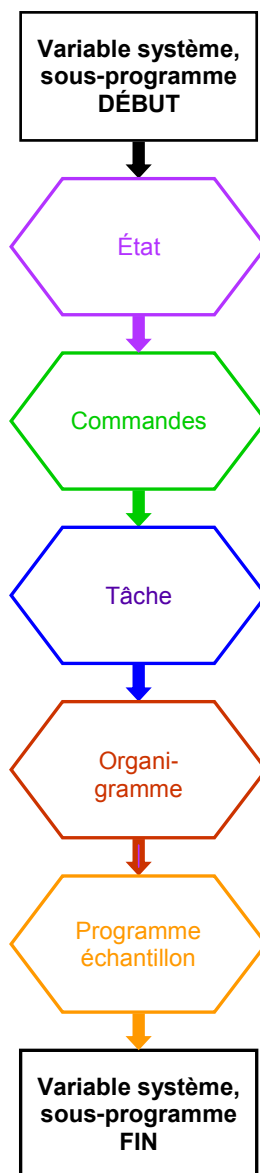
Contenu :

Explication des variables système

Explication des sous-programmes qui ne sont pas dans le même dossier

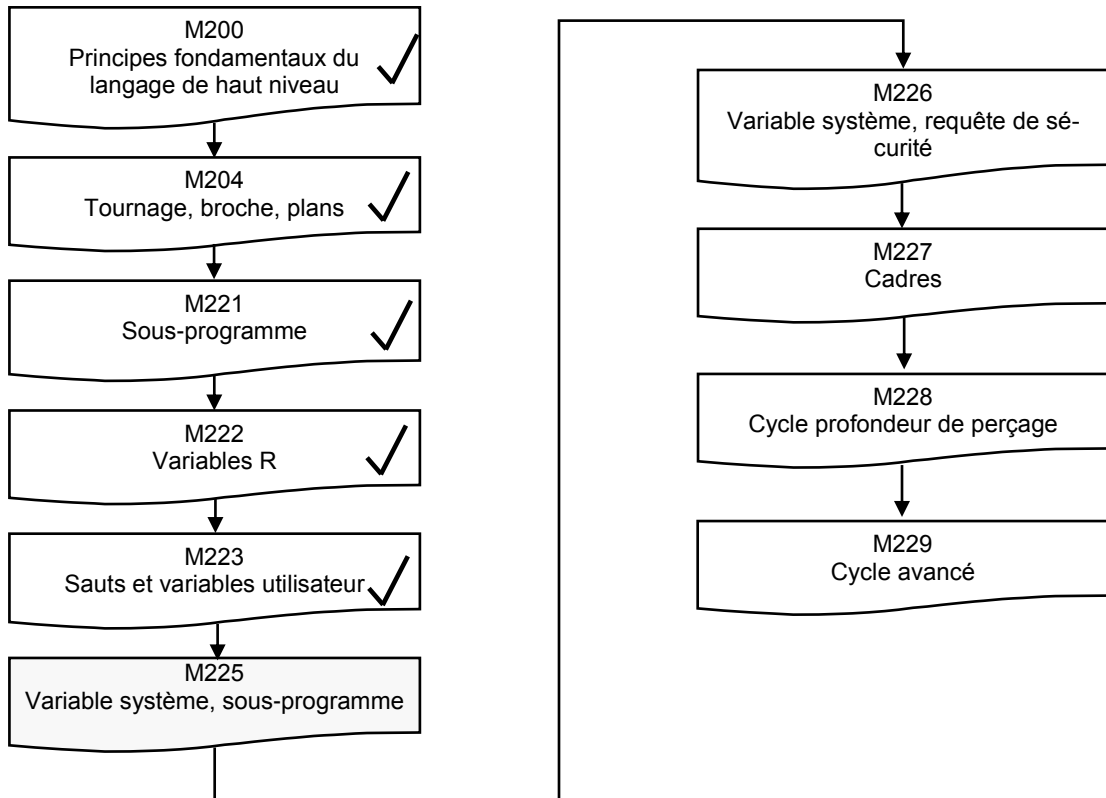
Programmation des variables utilisateur





Remarques

État :



Commandes :

Commande	Signification
\$AA_IW[MZ1]	Variable système
CALL	Appel
RET	Retour depuis le sous-programme

Variable système :

- ⇒ En plus des deux autres types de variable, la variable utilisateur (DEF) et la variable arithmétique (variable R), il existe également des variables système. Ces variables sont fournies par le fabricant de la machine et disposent d'une notation spéciale.
- ⇒ Les états de contrôle et de la machine peuvent être appelés via ces variables système et intégrés ensuite dans un programme NC.
- ⇒ Par exemple, des décalages d'origine, des positions d'axe et des décalages d'outil.
- ⇒ Ces variables peuvent être appelées, ainsi qu'écrites en partie.
- ⇒ Une liste des variables système et de leur signification est disponible sur Internet en utilisant la recherche "Siemens system variables", ainsi que dans le document PDF sur les variables système. Ce document peut être téléchargé librement.

Call :

- ⇒ Les sous-programmes peuvent être appelés indirectement avec cette commande. Indirectement signifie qu'ils ne sont pas dans le même dossier que le programme principal. L'instruction Call doit être écrite avec le nom du sous-programme et le type de fichier.
- ⇒ Le chemin entier du sous-programme peut être également saisi. De cette façon, le saut vers le sous-programme est plus rapide. Exemple : Call "_Name.spf"

RET :

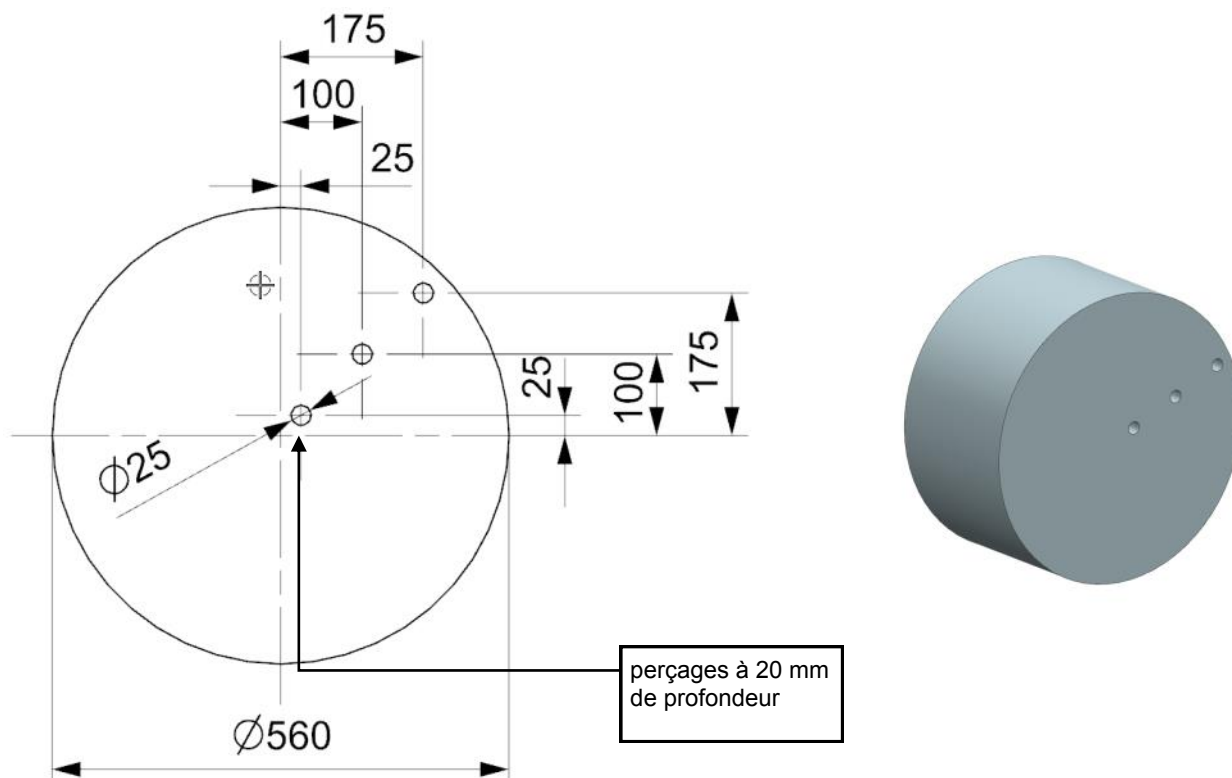
Cette commande peut être utilisée pour revenir d'un sous-programme, en ignorant l'API et sans interrompre le mode chemin continu actif.

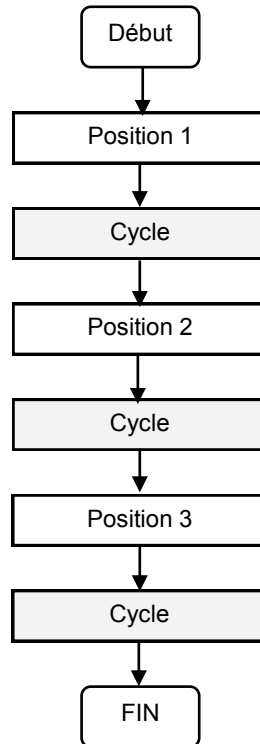
Remarques

Tâche :**Description de la tâche**

Trois trous doivent être percés dans une pièce.

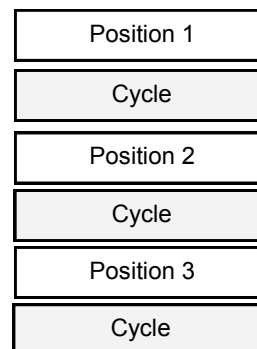
- ⇒ La position de rétractation, après l'opération d'usinage, doit être définie par une variable système et doit être la même pour les trois trous.
- ⇒ Le cycle avec la variable système doit être écrit dans un sous-programme.
- ⇒ Le mode chemin continu reste actif après la fin de l'appel du sous-programme.

Schéma :**Remarques**

Organigramme :Programme échantillon :**Programme principal :**

```

N10 G17 G60 G90 G94
N20 G54
N30 WORKPIECE(,,,"CYLINDER",192,0,-120,-100,560)
N40 G18 ;TURNINGPLANE
N50 SPOS=0 ;SPINDLEPOSITIONING
N60 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N70 T="SOLID DRILL25"
N80 S3=1000 M3=3
N90 TRANSMIT ;TRANSFORM MILLING INTO TURNING
N100 DIAMOF ;PROGRAMMING IN RADIUS
N110 G17 ;MILLINGPLANE
N120 F140
N130 G0 Z5
N140 G0 X25 Y25
N150 Call "sub_1.spf"
N160 G0 X100 Y100
N170 G0 Z5
N180 Call "sub_1.spf"
N190 G0 Z5
N200 G0 X175 Y175
N210 Call "sub_1.spf"
N220 G0 X100 Y100 Z100
N230 M30
  
```



Remarques

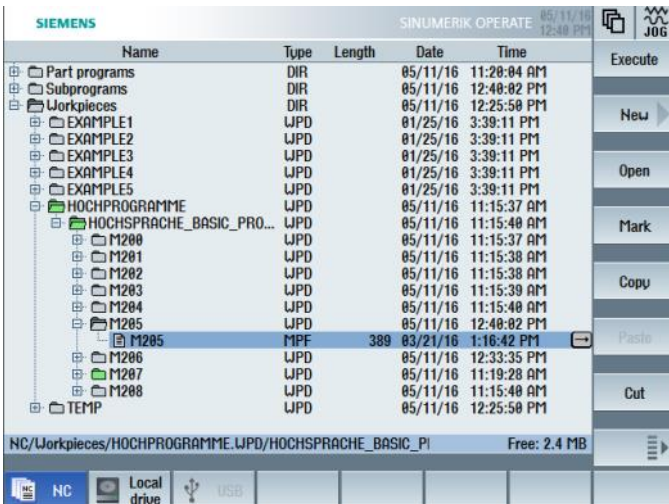
Sous-programme de cycle :

```

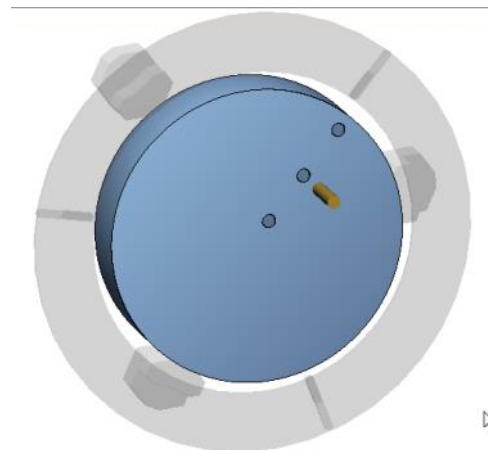
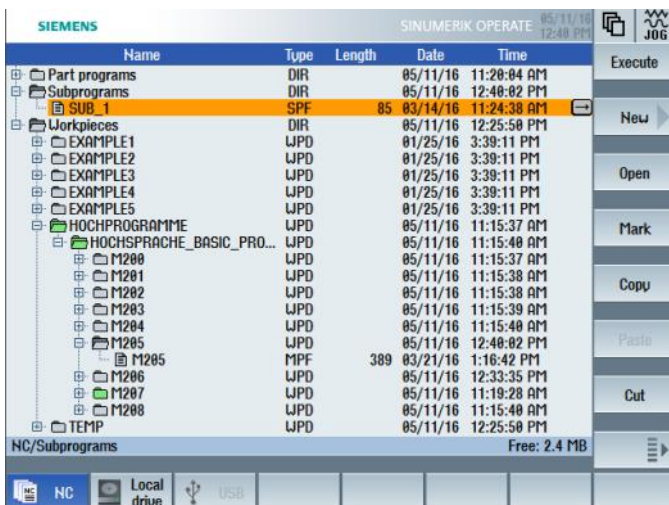
DEF real _Merk1      Definition of the user variable
N110_Merk1=$AA_IW[Z] Current position of the Z axis is linked to the user variable
N120 G1 Z-27
N130 Z=_MERK1       The system variable is approached
Ret                 Quick jump to main program
    
```

Remarque : La désignation de l'axe Z (\$AA_IW[Z]) doit correspondre au nom de l'axe de la machine.

Stockage du programme principal



Stockage du sous-programme de cycle :



la position de rétractation est toujours assurée avec ce cycle. Un espacement de sécurité a été programmé.

Remarques

Remarques

Description du module :

Le travail avec les variables système du module M 205 se poursuit dans ce module. Celui-ci traite des requêtes sur le type d'outil avec les variables appropriées.
Une boucle de sécurité est également intégrée pour le cas d'un type d'outil incorrect.

Cas d'utilisation :

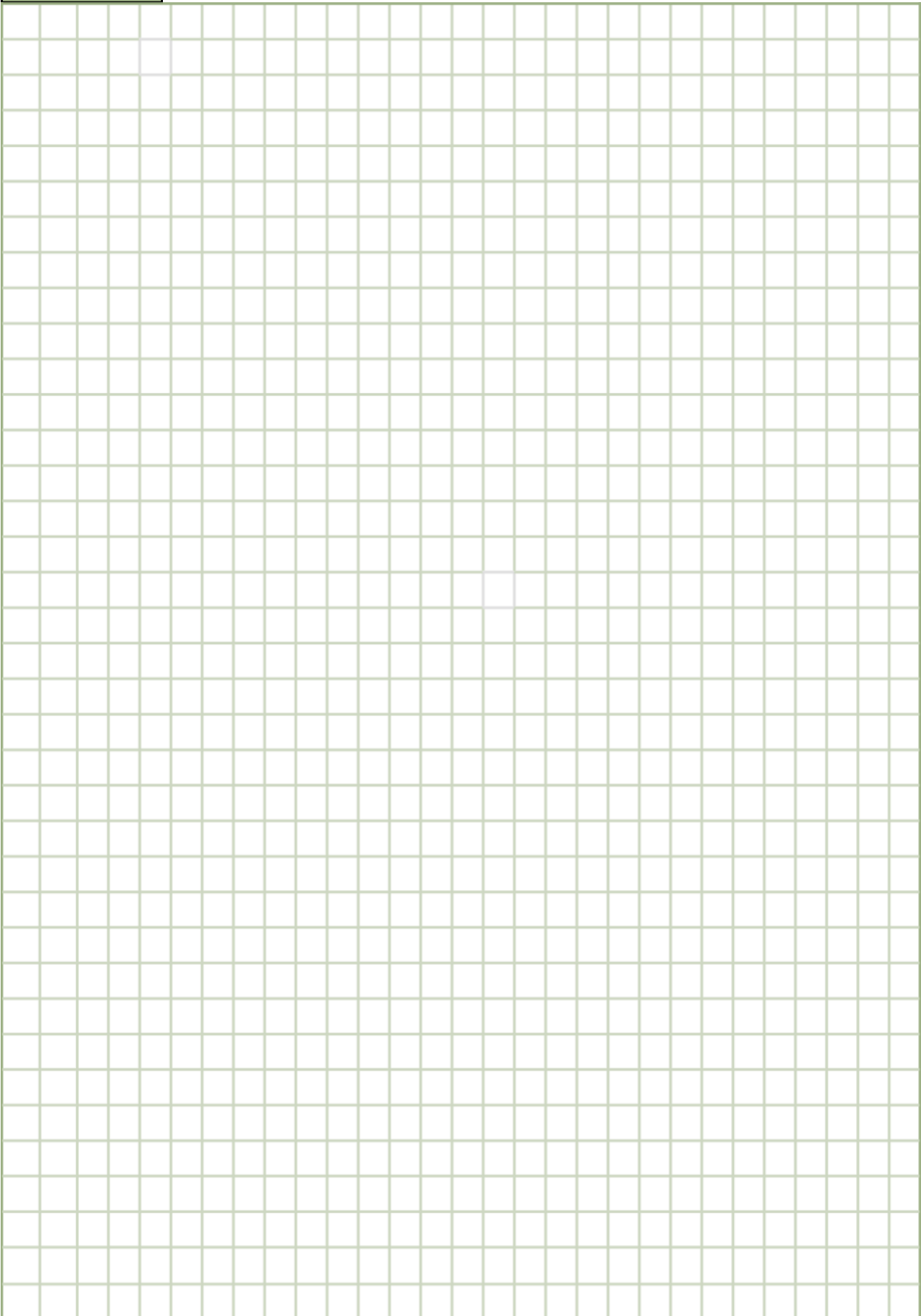
Après avoir complété le module, vous connaîtrez mieux les variables système et vous serez capable d'intégrer une boucle de sécurité dans un programme NC, de façon à ce que le programme ne puisse pas démarrer dans des conditions erronées.

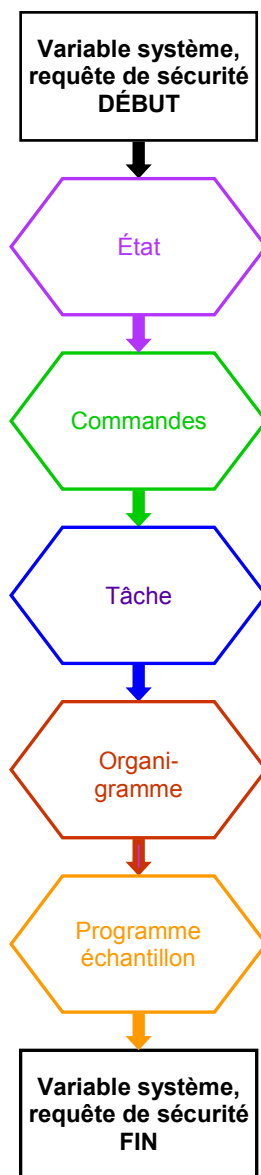
Contenu :

Explication des types d'outil

Explication des variables système utilisées

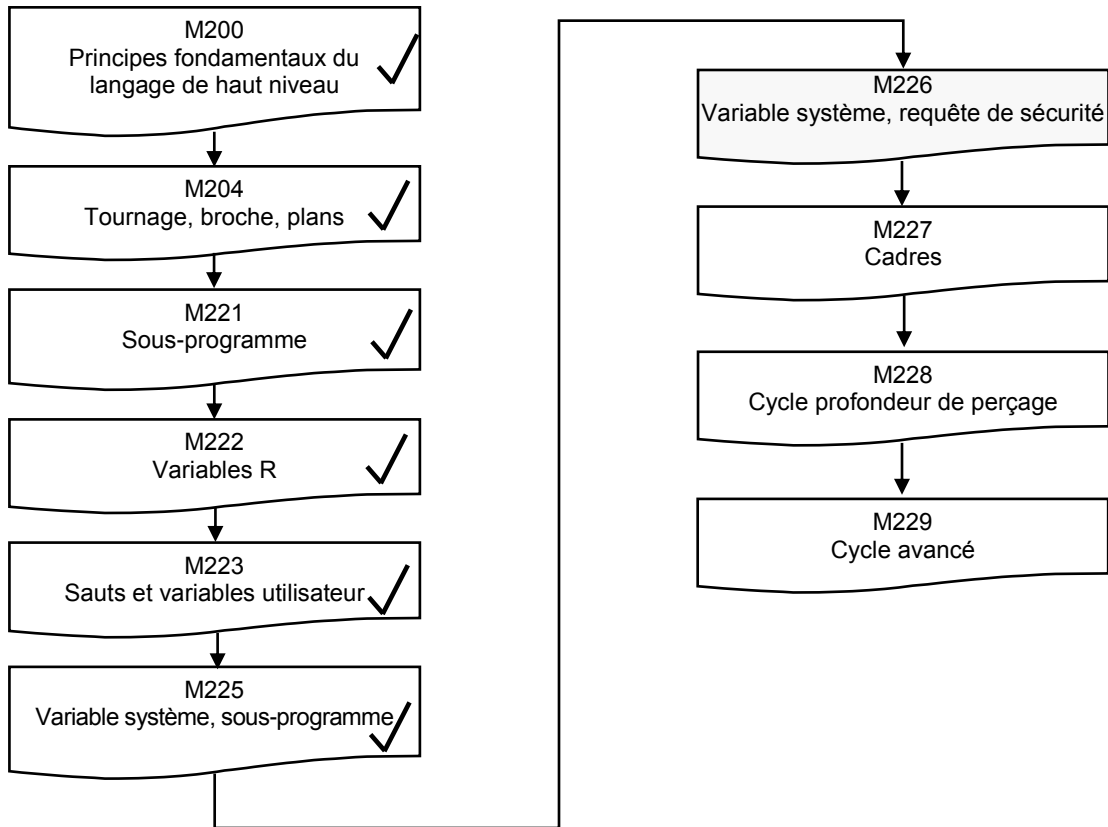
Programmation d'une boucle de sécurité qui se traduit par un "RESET" inconditionnel





Remarques

État :



Commande :

<u>Commande</u>	<u>Signification</u>
\$TC_DP[N]	Type d'outil
\$P_TOOLNO	Numéro d'outil actif
\$P_TOOL	Tranchant actif

\$TC_DP[N] :

Le type d'outil se définit par cette variable système.
 Dans SINUMERIK, les outils sont répartis entre différents types d'outil. À chaque type d'outil est attribué un nombre à 3 chiffres.
 Le premier chiffre attribue le type d'outil à un groupe d'outils avec la technologie appropriée.

Type d'outil	Groupe d'outils
1	Découpes
2	Forets
3	Réservé
4	Outils de meulage
5	Outils de tournage
6	Réservé
7	Outils spéciaux tels que fraise à trancher

Remarques

Outils de fraisage :

Le groupe 1 "Fraises" comprend les types d'outil suivants :

- 100 Outil de fraisage conforme à la norme CLDATA (Cutter Location Data)
- 110 Fraise à bille (fraise cylindrique d'enfonçage)
- 111 Fraise à bille (fraise conique d'enfonçage)
- 120 Fraise à queue (sans coin arrondi)
- 121 Fraise à queue (avec coin arrondi)
- 130 Fraise à tête d'angle (sans coin arrondi)
- 131 Fraise à tête d'angle (avec coin arrondi)
- 140 Outil à dresser les faces
- 145 Fraise pour filets courts
- 150 Fraise à surfacer en bout
- 151 Scie
- 155 Fraise à tailler les roues coniques (sans coin arrondi)
- 156 Fraise à tailler les roues coniques (avec coin arrondi)
- 157 Fraise conique d'enfonçage
- 160 Fraise de perçage et de filetage

Forets :

Le groupe 2 "Forets" comprend les types d'outil suivants :

- 200 Foret hélicoïdal
- 205 Foret
- 210 Barre d'alésage
- 220 Foret à centrer
- 230 Foret aléteur
- 231 Foret à lamer
- 240 Taraud normal
- 241 Taraud fin
- 242 Taraud Whitworth
- 250 Alésoir

Outils de tournage :

- 500 Outil d'ébauche
- 510 Outil de finition
- 520 Fraise à filer
- 530 Outil à saigner
- 540 Outil à fileter
- 550 Outil circulaire / outil de mortaisage (TOOLMAN)
- 560 Foret rotatif (ECOCUT)
- 580 Sonde avec paramètre de position pour tranchant

Seuls les groupes d'outil les plus importants pour le tournage et le fraisage sont répertoriés dans ce module.

Remarques

Le numéro de l'outil est visible dans SINUMERIK Operate lorsque le symbole d'outil "TYPE" est sélectionné dans le gestionnaire d'outil. Il affiche une liste avec les numéros de type des outils possibles.

Loc.	MT L.O.	Type	Tool name	ST	D	Length X	Length Z	∅	N
1		CUTTER_6		1	1	106.328	100.000	6.000	2
2		3D_PROBE		1	1	100.000	0.000	6.788	
3		ROUGHING_TOOL_55		1	1	120.000	0.000	0.200	95.0 80 1
4		PLUNGE CUTTER		1	1	100.000	0.000	0.100	3.000
5		DRILL_8_5		1	1	0.000	100.000	8.500	118.0
6		CENTERDRILL		1	1	0.000	100.000	12.000	90.0
7		TAP_10		1	1	0.000	100.000	10.000	1.000
8									
9		SOLID DRILL25		1	1	0.000	100.000	25.000	
10									
11									
12									

Type	Identifier	Tool position
500	- Roughing tool	[Icons]
510	- Finishing tool	[Icons]
520	- Plunge cutter	[Icons]
540	- Threading tool	[Icons]
550	- Button tool	[Icons]
560	- Rotary drill	[Icons]
580	- 3D turning probe	[Icons]
730	- Stop	[Icons]
120	- End mill	[Icons]
140	- Facing tool	[Icons]
150	- Side mill	[Icons]
200	- Twist drill	[Icons]
240	- Tap	[Icons]
	Multitool	[Icons]

\$P_TOOLNO

Cette variable système décrit le numéro de l'outil actif, c'est-à-dire l'outil actif dans le NC :

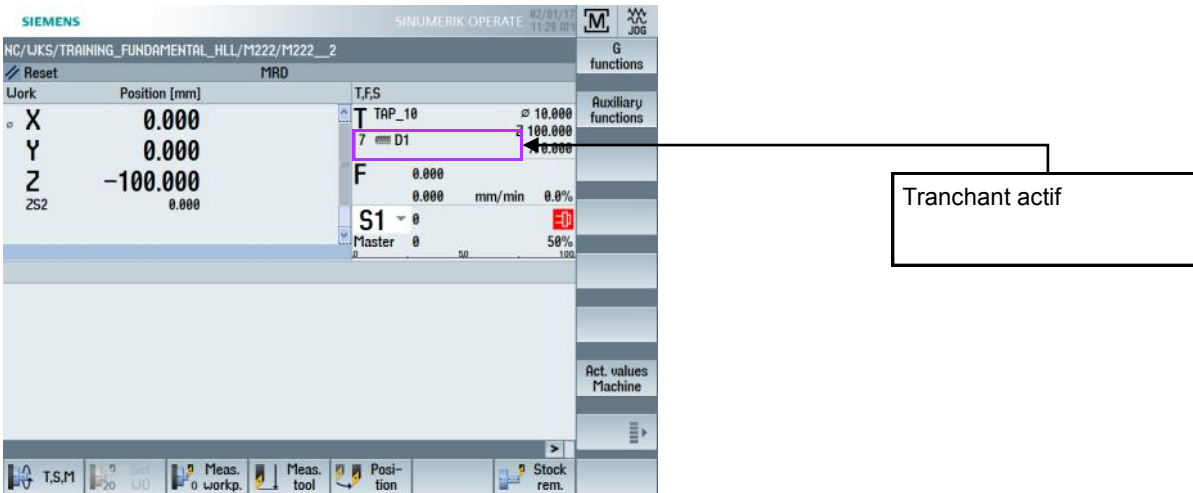
Work	Position [mm]	T,F,S
X	0.000	T TAP_10 10.000
Y	0.000	7 D1 100.000
Z	-100.000	X 0.000
ZS2	0.000	F 0.000
		0.000 mm/min 0.0%
		S1 0
		Master 0 50%

Outil actif NC

Remarques

\$P_Tool

Cette variable système décrit le tranchant actif et ses décalages actifs.



The screenshot shows the Siemens SINUMERIK OPERATE interface. The 'T.F.S.' (Tool Functions) section is highlighted with a purple box. The '7 D1' entry is highlighted with a white box, and a callout box labeled 'Tranchant actif' points to it. The interface also displays position data (X, Y, Z) and various machine parameters.

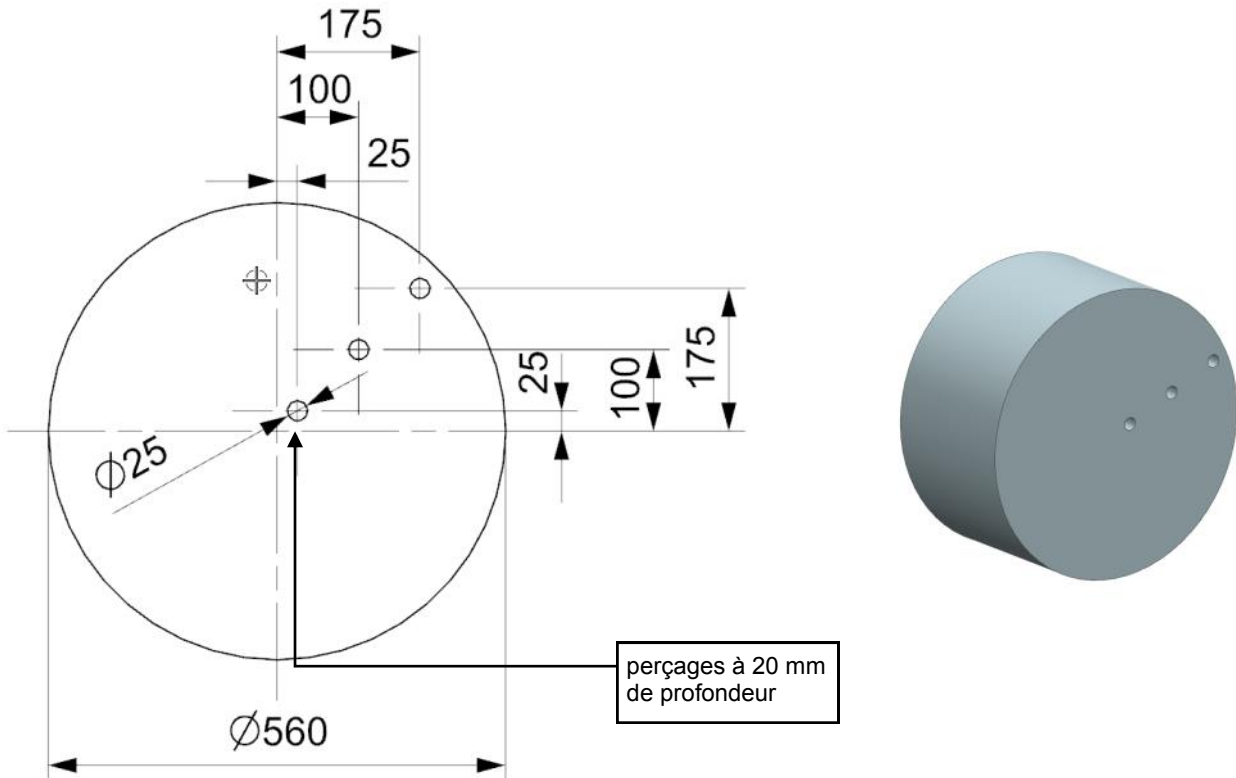
Tâche :**Description de la tâche**

La base de ce module est constituée de la pièce du module M205.

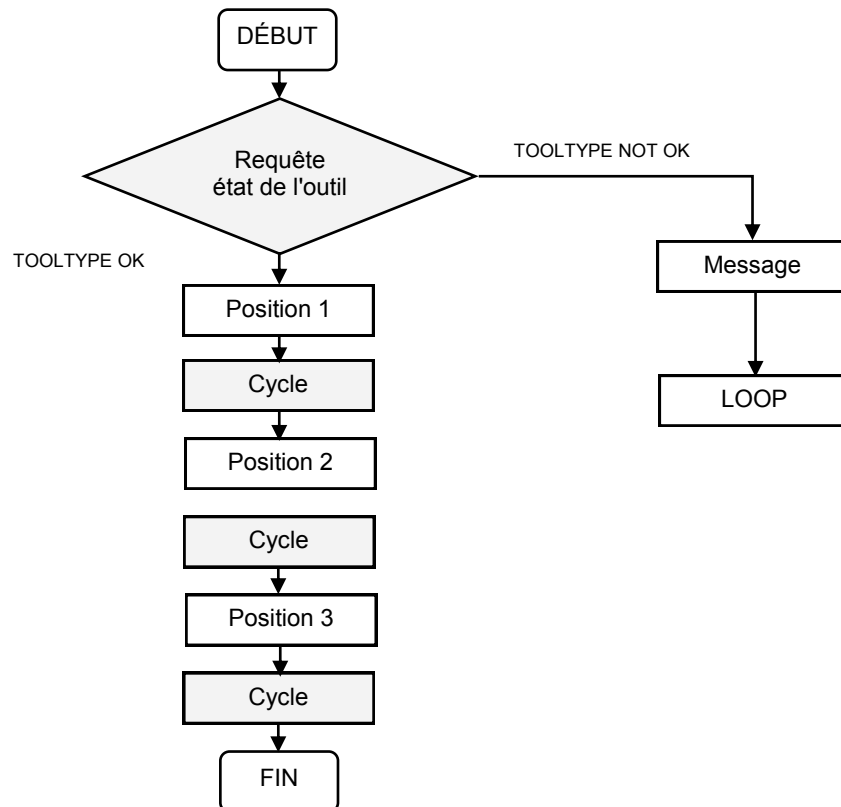
- ⇒ Une variable entière nommée "TOOLTYPE" doit être définie.
- ⇒ Le type d'outil, l'outil actuel dans le NC et le tranchant actuel sélectionné doivent être appelés via cette variable.
- ⇒ Si le type d'outil pour l'usinage n'est pas correct, le programme sautera dans une boucle de laquelle l'opérateur ne pourra sortir qu'en appuyant sur "Program reset". Un message s'affiche pour l'opérateur si un outil incorrect est actif.

Remarques

Schéma :



Organigramme :



Remarques

Programme échantillon :

Programme principal :

```

;_TOOLTYPE= VARIABLE FOR TOOL TYPE WITH CURRENT
;TOOL NUMBER AND CUTTING EDGE NUMBER
N10 DEF INT _TOOLTYPE
N20 G17 G60 G90 G94
N30 G54
N40 WORKPIECE(,,, "CYLINDER",192,0,-120,-100,560)
N50 G18 ;TURNINGPLANE
N60 SPOS=0 ;SPINDLEPOSITIONING
N70 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N80 T="SOLID DRILL25"
N90 S3=1000 M3=3
N100 TRANSMIT ;TRANSFORM MILLING INTO TURNING
N110 DIAMOF ;PROGRAMMING IN RADIUS
N120 G17 ;MILLINGPLANE
N130 F140
N140 _TOOLTYPE = $TC_DP1[$P_TOOLNO,$P_TOOL]
N150 IF (_TOOLTYPE <200) OR (_TOOLTYPE >299) GOTOF _ERROR1
N160 G17 G40 G60 G71 G90
N170 G0 Z5
N180 G0 X25 Y25
N190 Call "sub_1.spf"
N200 G0 X100 Y100
N210 G0 Z5
N220 Call "sub_1.spf"
N230 G0 Z5
N240 G0 X175 Y175
N250 Call "sub_1.spf"
    
```



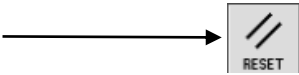
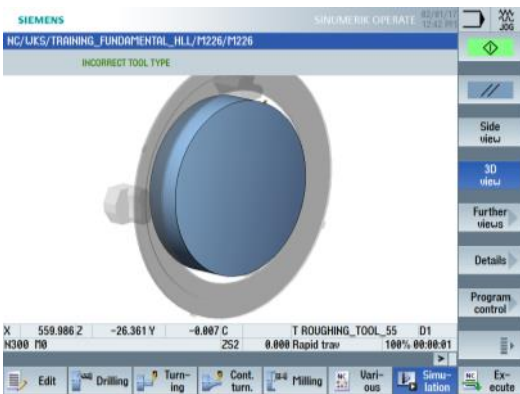
```

N260 G0 X100 Y100 Z100
N270 M30

N280 _ERROR1:
N290 MSG("INCORRECT TOOL TYPE")
N300 M0
N310 GOTOB _ERROR1
    
```

Message

LOOP



Tâche additionnelle :

⇒ Ce programme ne peut être exécuté que si un foret solide est sélectionné.

Remarques

Remarques

Description du module :

La lecture et l'écriture des cadres sont décrites dans ce module.

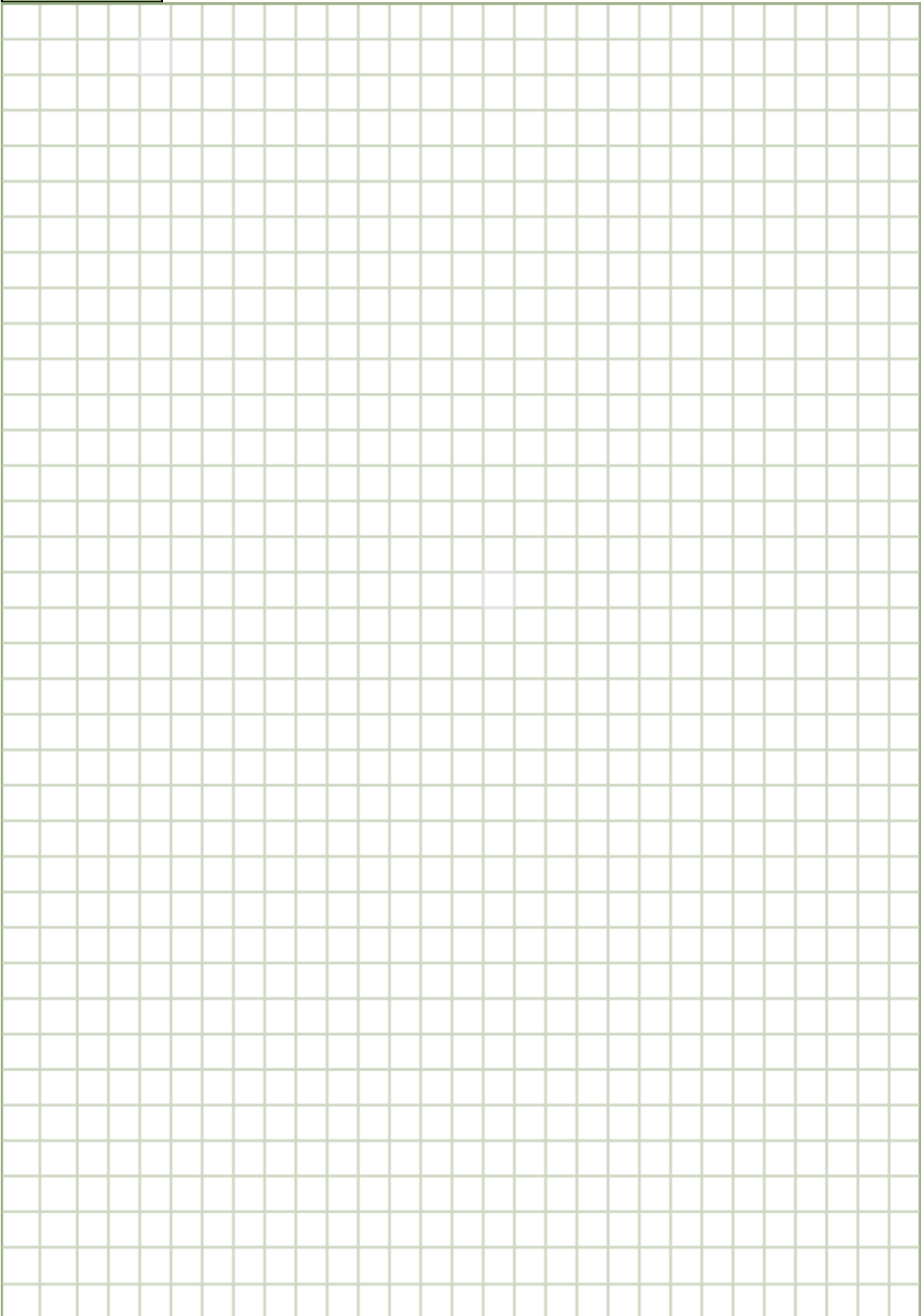
Cas d'utilisation :

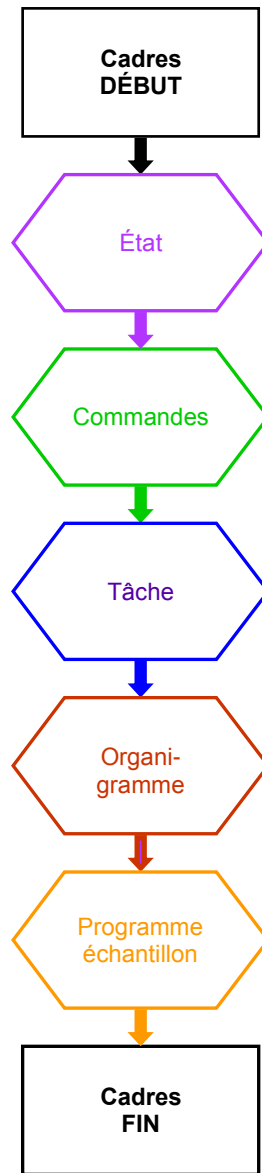
Après avoir complété ce module, vous pourrez programmer un cadre utilisateur pour le perçage et vous pourrez lire et écrire des cadres.

Contenu :

Structure des cadres

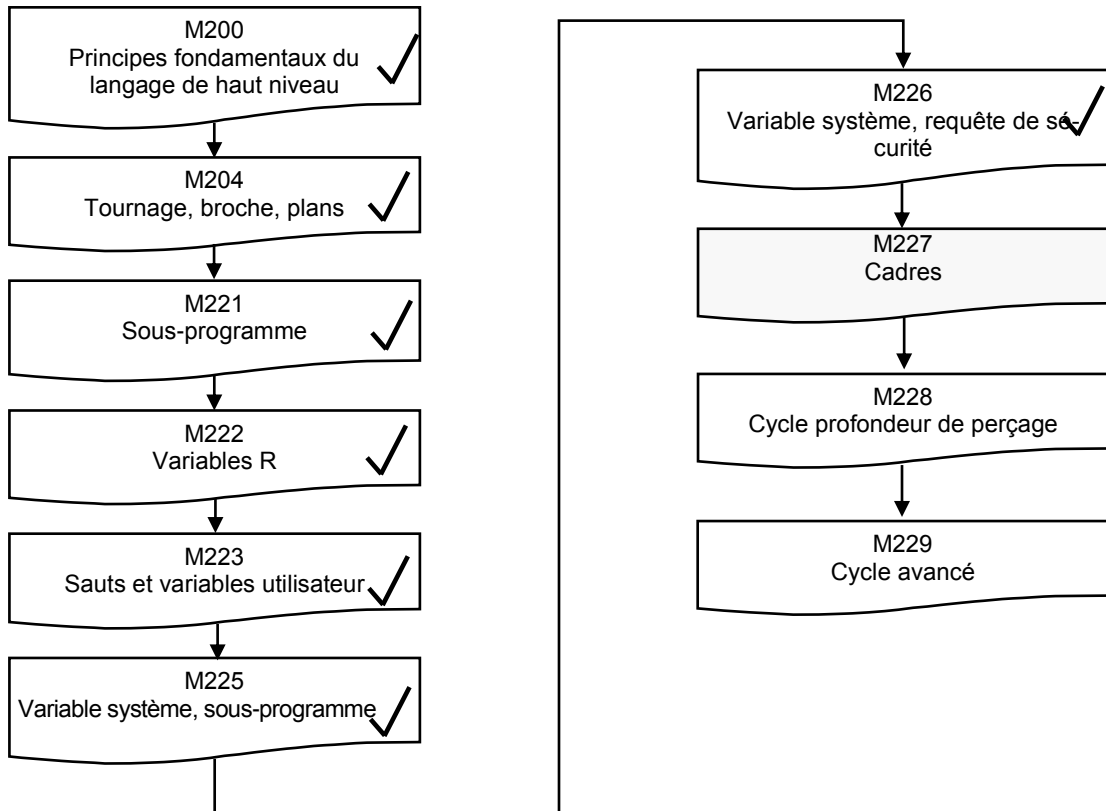
Explication des commandes et des variables utilisées





Remarques

État :



Commandes :

Ce module ne se réfère qu'au cadre d'un décalage d'origine réglable, par exemple G54, \$P_UIFR[1]

<u>Commande</u>	<u>Signification</u>
\$P_GG[n]	Lire la fonction G active du groupe de fonctions G
\$P_UIFR	Cadre réglable actif
CTRANS	Décalage brut
CFINE	Décalage fin

\$P_GG[n] :


Les fonctions G sont réparties dans des groupes de fonctions. Cela empêche les commandes d'un groupe d'être appelées deux fois dans un même bloc.

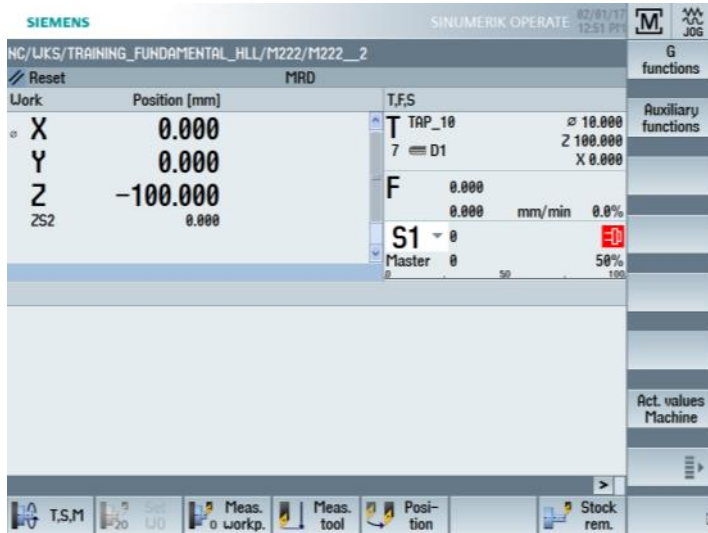
Remarques

Exemple :

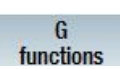
Les commandes G G0, G1, G2 et G3 appartiennent au groupe de fonctions 1. Logiquement, elles ne peuvent pas être insérées dans un bloc, car elles font partie du même groupe de fonctions. (Manuel des principes fondamentaux de SINUMERIK, chapitre 12, tableaux et liste d'instructions).

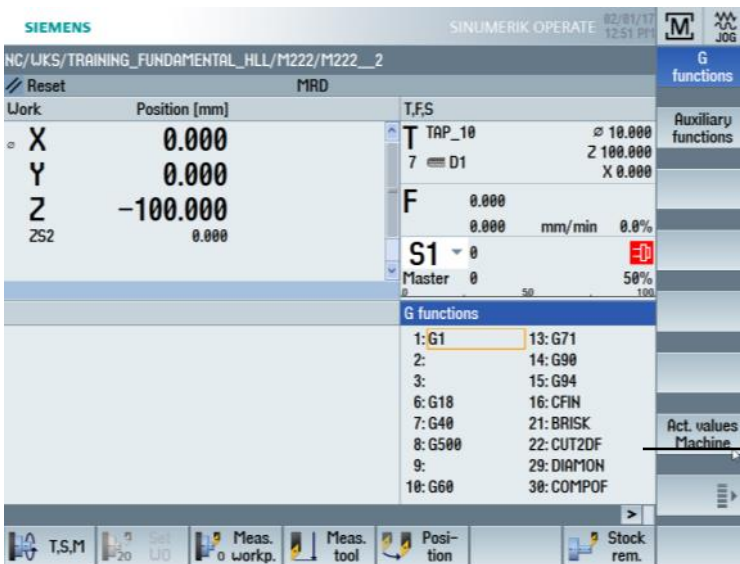
Il est toujours possible à n'importe quel moment d'afficher les groupes de code G actifs et leur contenu dans SINUMERIK Operate.

Dans la zone d'opération Machine,  il existe deux options :



1ère option :

En appuyant sur la touche programmable , les groupes de fonctions G les plus importants associés à la commande active s'affichent.



G functions	
1: G1	13: G71
2:	14: G90
3:	15: G94
6: G18	16: CFIN
8: G500	21: BRISK
9:	22: CUT2DF
10: G60	29: DIAMON
	30: COMPOF

Remarques

2e option

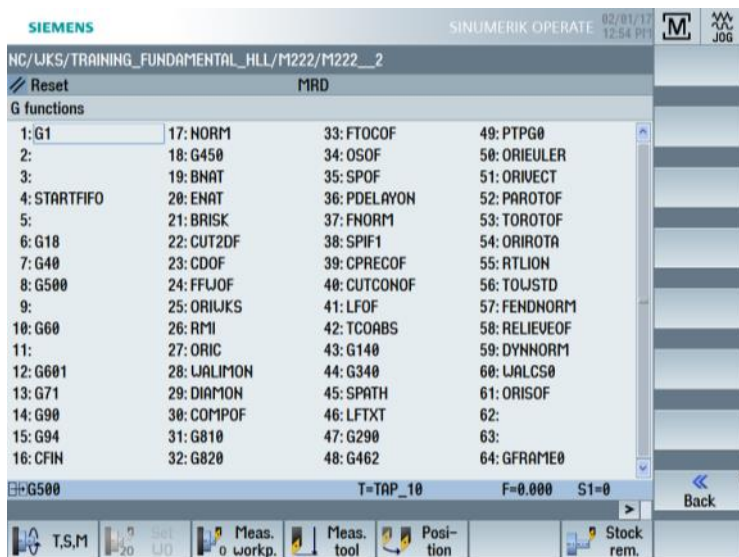
En développant la barre des touches programmables



et en appuyant sur la touche programmable

All G
functions

tous les groupes de fonctions G actifs s'affichent avec la commande active.



Cette vue peut être également affichée pendant l'usinage, de façon à ce que vous puissiez observer quelle est la commande active dans un groupe pendant l'exécution du programme.

\$P_UIFR :

Cette variable système permet d'accéder directement aux valeurs des décalage d'origine réglables.

\$P_UIFR[N]

Le nombre entre crochets indique le décalage d'origine concerné.

La commande qui décrit ce qu'il y a à faire avec les valeurs du décalage d'origine se place après la variable système.

CTTRANS :

Un décalage d'origine (décalage brut) peut être écrit avec cette fonction cadre.

L'axe est d'abord défini, puis la valeur dans le paramètre suivant.

La commande CTRANS ne peut pas être utilisée seule, uniquement en conjonction avec une variable système.

Remarque : Si un seul axe bénéficie d'une valeur, tous les autres axes sont réglés sur zéro.

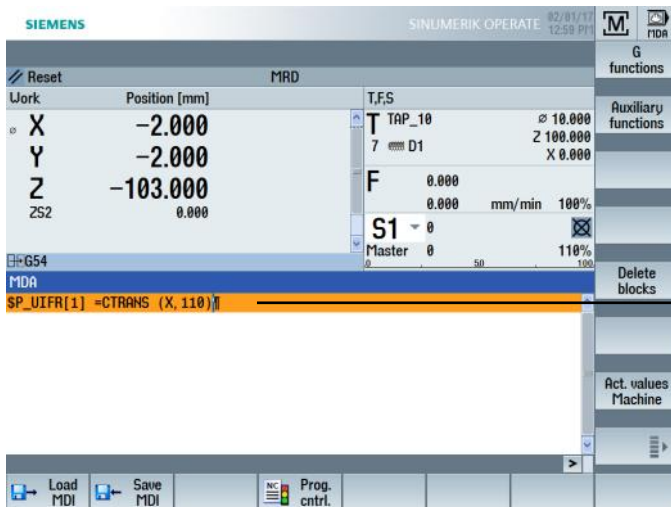
Exemple :

Valeurs actives dans le décalage d'origine G54


Work offset - active [mm]						
	X	Y	Z	ZS2	C	
Machine act value	0.000	0.000	0.000	0.000	0.000	0.000
G54	1.000	2.000	3.000	0.000	0.000	0.000

Remarques

Saisie dans MDI :



SP_UIFR[1]
 Le nombre entre crochets concerne G54
=CTTRANS (X,110)
 L'axe X reçoit la valeur 110

En appuyant sur , la valeur de "X" est prise dans le décalage d'origine G54. Les valeurs de tous les autres axes sont réglées sur zéro.

Work offset - active [mm]						
	X	Y	Z	ZS2	C	
Machine act value	0.000	0.000	0.000	0.000	0.000	0.000
G54	110.000	0.000	0.000	0.000	0.000	0.000
Total W0	110.000	0.000	0.000	0.000	0.000	0.000

CFINE :

Le comportement avec CFINE est identique à celui de CTRANS. Toutes les valeurs actives du décalage fin sont réglées sur zéro si un seul axe bénéficie d'une valeur. Voir l'exemple ci-dessus.

CFINE est utilisé lorsque vous ne voulez pas modifier les valeurs dans les décalages d'origine (décalages bruts).

Vous pouvez utiliser des niveaux d'accès pour qu'un opérateur, par exemple, ne puisse accéder qu'aux valeurs du décalage fin.

La taille maximale du décalage fin peut être également spécifiée.

Remarque :

Les valeurs de CFINE sont toujours actives pour le décalage d'origine et sont incluses dans le calcul des axes respectifs.

Il est essentiel que le décalage fin soit également pris en compte pour le décalage d'origine.

Si toutes les valeurs d'un cadre ne doivent pas être supprimées, mais seules les valeurs d'un axe particulier, il suffit d'utiliser la commande TR dans le cadre.

Exemple : \$P_UIFR[1,X,TR]=25

Dans le cadre G54, seule la valeur de l'axe X est modifiée avec 25. Toutes les autres valeurs restent inchangées

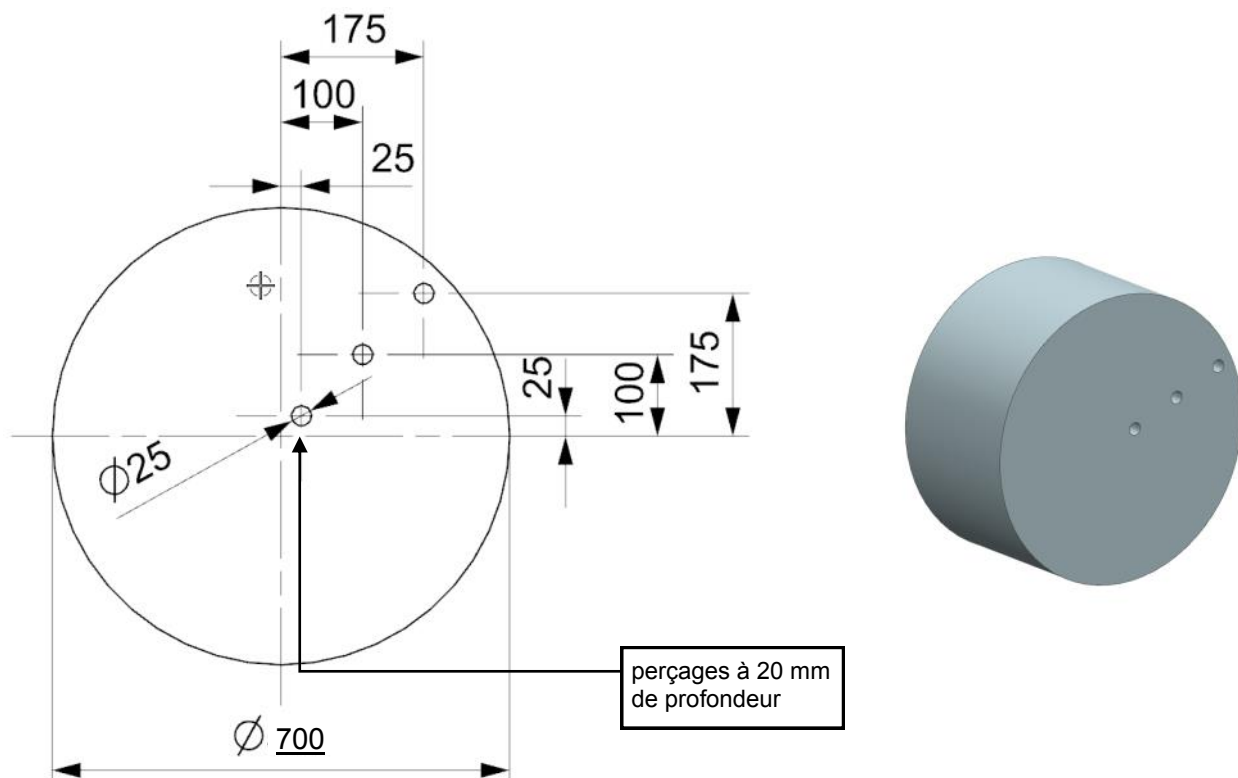
Remarques

Tâche :**Description de la tâche**

La base de ce module est la pièce du module M206.

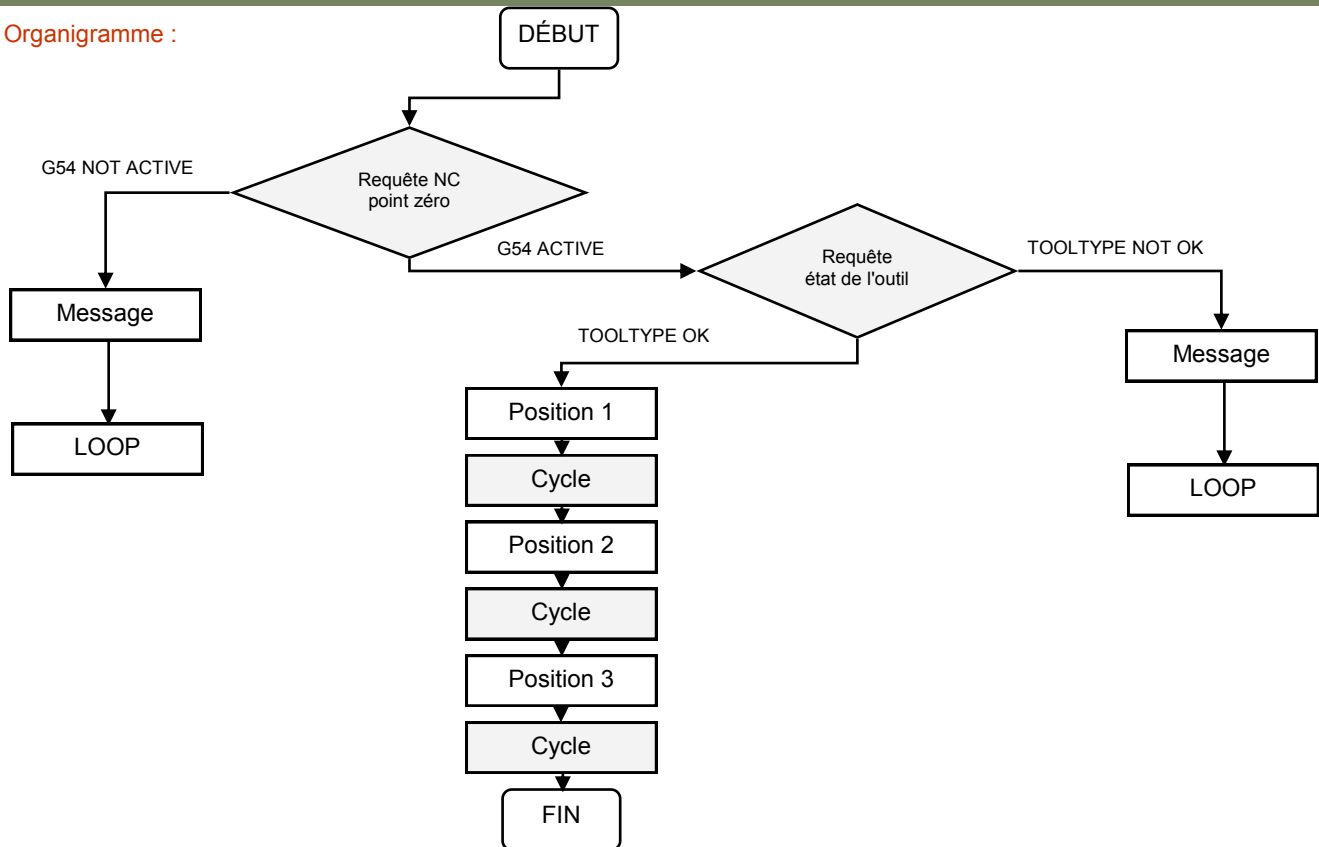
Hypothèse : La pièce est fixée via un système de fixation permanent au point zéro défini. Les coordonnées pour ce faire sont prédéfinies et écrites dans un point zéro.

- ⇒ Le programme doit être configuré de façon à ce que le groupe de code G soit appelé et que le décalage d'origine G54 soit actif. L'opérateur ne doit pas modifier les décalages d'origine.
- ⇒ Si le décalage d'origine actif n'est pas G54, le programme s'exécute en boucle et ne peut être interrompu qu'en appuyant sur "RESET".
- ⇒ La requête des variables système pour le cadre du système et la description doit prendre en compte le décalage fin, en plus du décalage brut.

Schéma :

Remarques

Organigramme :



Programme échantillon :

Programme principal

```

;_TOOLTYPE= VARIABLE FOR TOOL WITH CURRENT
;TOOLNUMBER AND CUTTING EDGE NUMBER
N10 DEF INT _TOOLTYPE
N20 DEF REAL _POSITION_X
N30 DEF REAL _POSITION_Y
N40 DEF REAL _POSITION_Z

;-----
;POSITION OF ZERO POINT CLAMPING SYSTEM
N50 _POSITION_X=1
N60 _POSITION_Y=1
N70 _POSITION_Z=0
;-----

N90 R10 = $P_GG[8]
N100 IF $P_GG[8] <> 2 GOTOF _ERROR2
N110 $P_UIFR[$P_GG[1]]=CTRANS (X, _POSITION_X, Y, _POSITION_Y, Z,
 _POSITION_Z):CFINE(X,0,Y,0,Z,0)
;_TOOLTYPE= VARIABLE FOR TOOL WITH CURRENT
N120 G17 G60 G90 G94
N130 WORKPIECE(,,,"CYLINDER",192,0,-120,-100,700)
N140 G18 ;TURNINGPLANE
N150 SPOS=0 ;SPINDLEPOSITIONING
N160 SETMS(3) ;DRIVEN TOOL IS MASTERPINDLE
N170 T="SOLID DRILL25"
;T="ROUGHING_TOOL_55"
N180 S3=1000 M3=3
N190 TRANSMIT ;TRANSFORM MILLING INTO TURNING
  
```

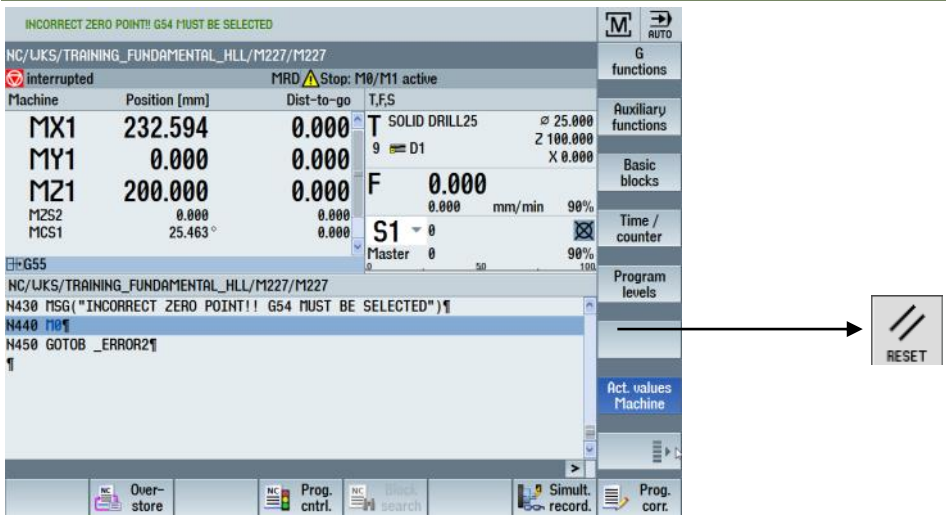


Remarques

```

N200 DIAMOF                ;PROGRAMMING IN RADIUS
N210 G17                  ;MILLINGPLANE
N220 F140
N230 _TOOLTYPE = $TC_DP1[$P_TOOLNO,$P_TOOL]
N240 IF (_TOOLTYPE <200) OR (_TOOLTYPE >299) GOTOF _ERROR1
N250 G17 G40 G60 G71 G90
N260 G0 Z5
N270 G0 X25 Y25
N280 Call "sub_1.spf"
N290 G0 X100 Y100
N300 G0 Z5
N310 Call "sub_1.spf"
N320 G0 Z5
N330 G0 X175 Y175
N340 Call "sub_1.spf"
N350 G0 X100 Y100 Z100
N360 M30

N370 _ERROR1:
N380 MSG("INCORRECT TOOL TYPE")
N390 M0
N400 GOTOB _ERROR1
    
```



Remarques

Description du module :

Le travail avec un cycle auto-programmé est décrit dans ce module.

Le cycle concerne le programme entier et il est appelé depuis le programme principal, avec le transfert des paramètres requis pour l'opération d'usinage.

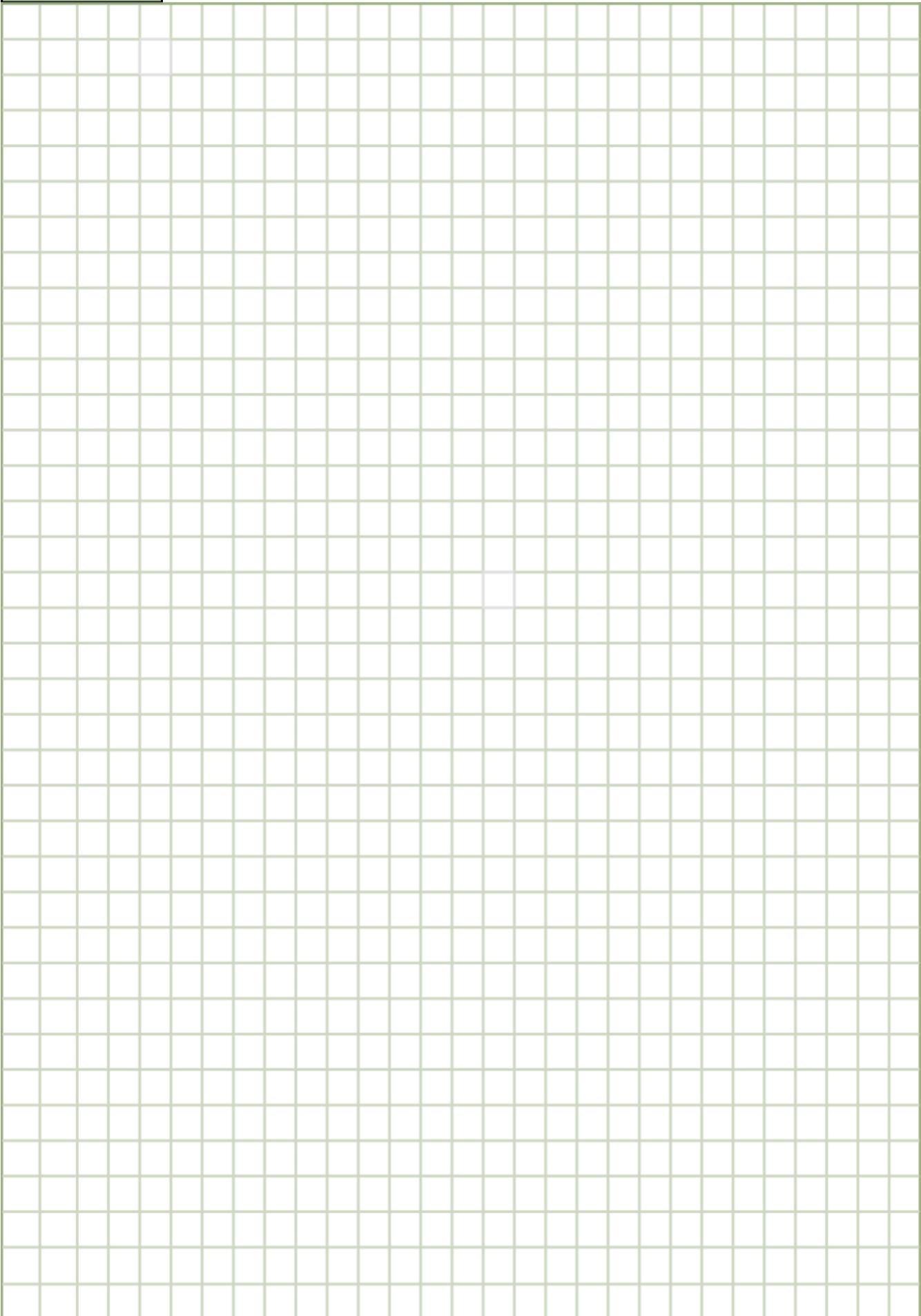
Cas d'utilisation :

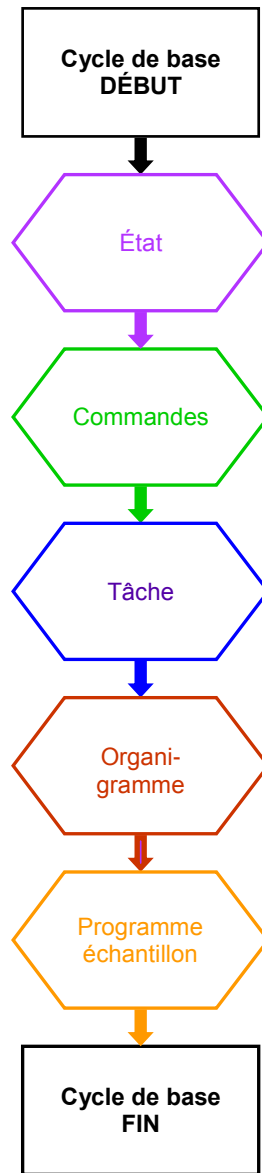
Après avoir complété ce module, vous serez en mesure d'écrire un cycle qui correspondra exactement à vos exigences et qui sera effectif dans tous les programmes. De cette façon, le temps de programmation peut être réduit et la production augmentée. Les erreurs de programmation sont également diminuées parce que le cycle contient déjà des sections de programme importantes.

Contenu :

Configuration et gestion d'un cycle qui soit efficace dans tous les programmes

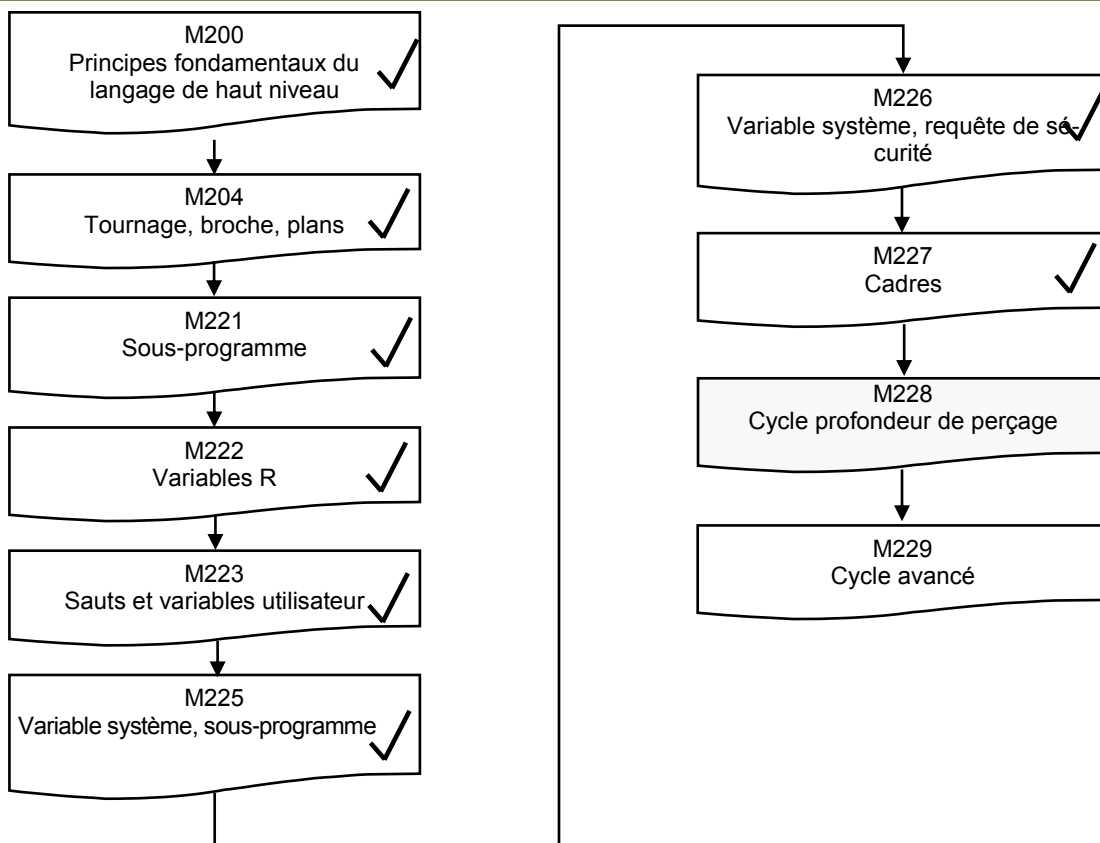
Écriture de votre propre cycle pour une profondeur de perçage





Remarques

État :



Commandes :

Commande	Signification
PROC	Première instruction d'un programme avec déclaration de variables pour récupérer les valeurs depuis le programme principal
DISPLOF	L'affichage du bloc actuel est supprimé
SAVE	Enregistrement des fonctions G modales
IF/ENDIF	Boucles de programmation pour vérifier la cohérence du contenu des variables transférées

PROC :

Cette commande est utilisée pour les sous-programmes dont les valeurs de paramètre doivent être remplacées par celles du programme appelant.

Exemple :

Le cycle MY_DRILLSPF est un cycle utilisateur auto-écrit. Le cycle doit être actif dans le programme principal, c'est-à-dire que les valeurs programmées et les paramètres doivent être insérés dans le programme principal.

Si un cycle utilisateur accepte cette condition, la première commande est toujours "PROC".

Tous les cycles d'origine Siemens commencent par cette commande.

Remarques

DISPLOF :

L'affichage du bloc actuel peut être supprimé avec cette commande.

Exemple : Si les cycles sont appelés dans un programme principal, les blocs NC individuels du cycle actuel MY_DRILL.SPF ne sont pas visibles. Ils sont supprimés. Les cycles sont généralement programmés de cette façon.

ENDIF/ELSE :

Ces commandes sont utilisées si la boucle de programmation contient un bloc de programme alternatif.

Si la condition IF est vraie, le bloc IF est exécuté.

Si la condition IF est fautive, le bloc de programme après la commande ELSE est exécuté.

La commande ENDIF marque la fin de la boucle IF. Le programme revient à l'état de la boucle.

SAVE :

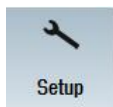
Avec la commande SAVE, les fonctions G modales et le cadre actif actuel avant l'appel du sous-programme sont enregistrés par le programme principal et actifs de nouveau après le retour.

Utilisation des cycles dans SINUMERIK :

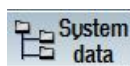
Pour Siemens, les cycles sont toujours des sous-programmes se terminant par ".SPF".

Leur philosophie est la suivante :

Dans la zone d'opération



à



il existe trois autres dossiers dans le dossier Cycles.

<ul style="list-style-type: none"> [-] Cycles <ul style="list-style-type: none"> [+] User cycles [+] Manufacturer cycles [+] Standard cycles 	DIR	06/29/16	9:17:50 AM
	DIR	05/24/16	2:24:51 PM
	DIR	07/04/16	12:18:29 PM

Standard cycles

Tous les cycles de SINUMERIK Operate se trouvent dans ce dossier. Ces cycles sont des sous-programmes qui peuvent être ouverts et modifiés. Toutefois, lorsque le contrôle est rechargé, les cycles reviennent à leur état d'origine.

Manufacturer cycles

Les cycles en provenance du fabricant sont placés dans ce dossier.

Les cycles modifiés sont actifs et stockés. Ils ne peuvent être modifiés que par le fabricant de la machine.

User cycles

Les cycles rédigés par l'opérateur sont stockés dans ce dossier.

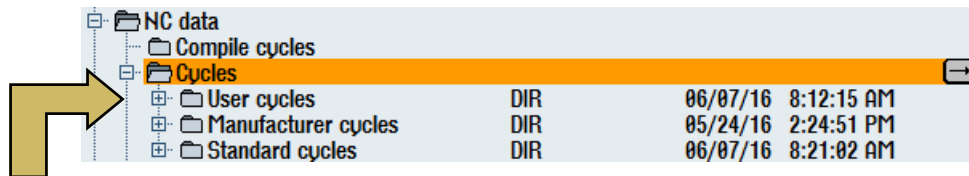
Les cycles modifiés sont actifs et stockés.

Remarques

Procédure :

Lorsqu'un cycle utilisateur est créé, il n'a pas besoin d'être entièrement écrit. Il est possible, par exemple, de modifier les cycles standard. Si vous souhaitez modifier un cycle existant, procédez comme suit :

Le cycle standard "Drilling depth" du dossier "Standard cycles" est programmé dans le dossier "User cycles".



Cycle de "Standard cycles" à "User cycles"

Le nom du cycle est conservé, les modifications sont effectuées et le contrôle est redémarré. Après le chargement, le cycle Drilling depth modifié est disponible.

Remarque : Il n'est pas recommandé de modifier les cycles standard mais, si vous devez le faire, soyez prudent.

La routine suivante s'applique lors de l'appel d'un cycle :

Le système vérifie si le cycle appelé se trouve bien dans le dossier "User cycles". Si c'est le cas, il est exécuté. Mais si le cycle appelé ne figure pas dans le dossier "User cycles", mais dans le dossier "Manufacturer cycles", il est exécuté.

Si le cycle appelé ne se trouve que dans le dossier "Standard cycles", il est utilisé.

Cette procédure garantit que le tout dernier cycle du programme soit **toujours** utilisé et que, par conséquent, les modifications soient activées.

Si vous écrivez un nouveau cycle, il sera stocké directement dans le dossier "User cycles".

Si un cycle est appelé en propre, l'opérateur n'aura aucune influence directe sur lui parce qu'en général, il n'est pas visible et que l'opérateur ne sait pas où le trouver.

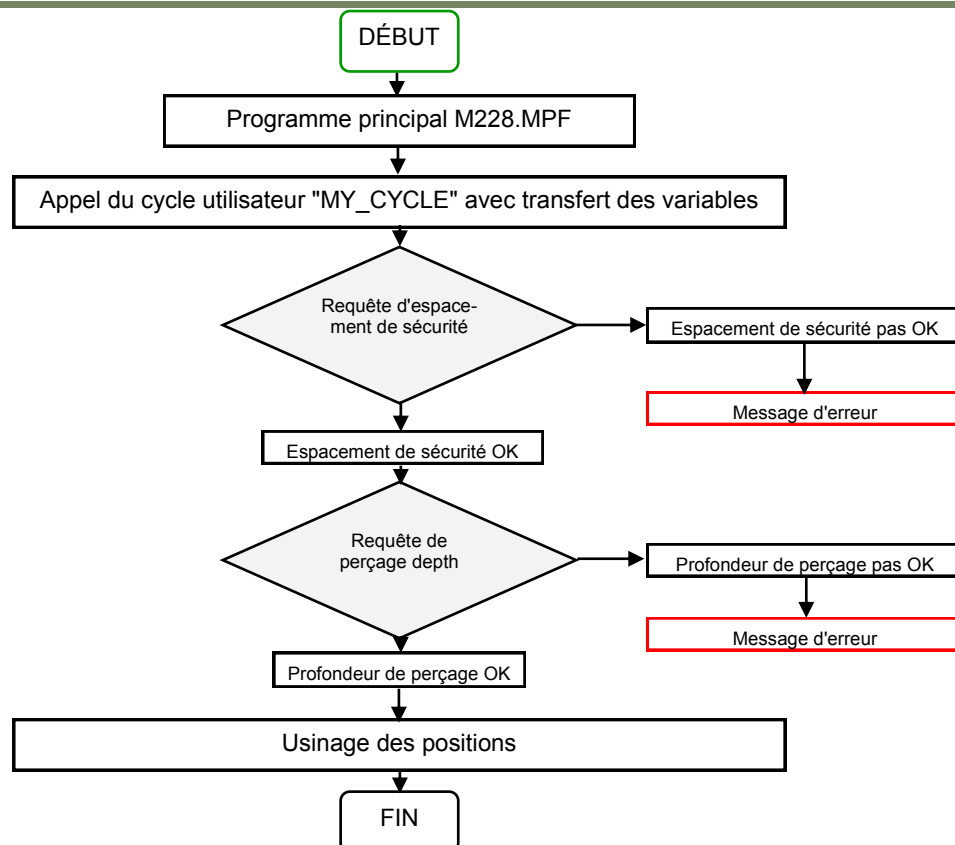
Lorsque le contrôle est rechargé, le cycle devient actif dans le dossier "User cycles".

Remarques

Tâche :

Un simple cycle de perçage nommé "My_cycle" doit être programmé et utilisé comme cycle utilisateur.

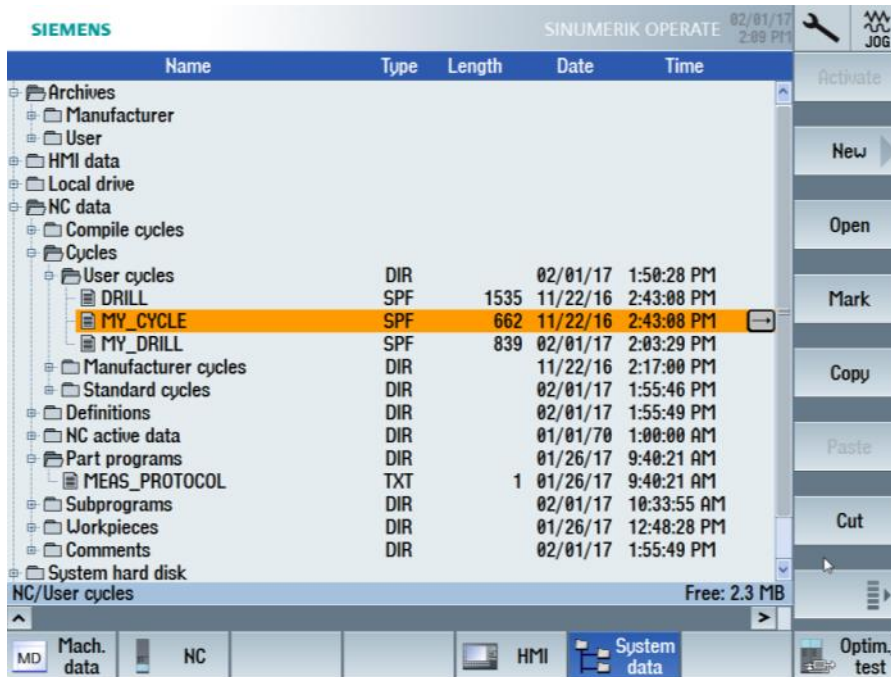
- ⇒ Le cycle doit être programmé dans le programme principal et être invisible pendant l'exécution du programme.
- ⇒ Le cycle récupère les valeurs dans le programme principal : Espacement de sécurité, profondeur de perçage, avance et rétractation.
- ⇒ Si un espacement de sécurité différent de zéro est transféré, un message d'erreur s'affiche.
- ⇒ Si la profondeur de perçage ne transmet pas une valeur négative, un message d'erreur s'affiche également.

Organigramme :

Remarques

Programme échantillon

Pour créer le cycle utilisateur MY_CYCLE.SPF, ouvrir le dossier **User cycles** et créer un fichier MY_CYCLE.SPF en appuyant sur la touche **New**.

**Programme principal 228.MPF**

```

;=====
;M228 MAIN PROGRAM FOR DRILLING WITH PARAMETER TRANSFER TO MY_CYCLE
;=====
;
;TRANSFER PARAMETERS TO THE MY_CYCLE
;-----
; SAFE           ;SAFETY CLEARANCE- POSTIVE NUMERIC VALUEt
;_DEPTH          ;DRILLING DEPTH- NEGATIVE NUMERIC VALUE
;_FEED           ;FEEDRATE MM/REV- POSTIVE NUMERIC VALUEt
;_RETR           ;RETRACTION AFTER DRILLING- POSTIVE NUMERIC VALUE
N10 G17 G64 G95 G90           ;Set environment
;
N20 G54                       ;Select 1st WO
;
N30 WORKPIECE(,,,"CYLINDER",64,0,-100,-80,700)
;
N40 T="SOLID DRILL25"
;
N50 G95 S2000 M03             ;SET TECHNOLOGY
N60 MCALL MY_CYCLE(1,-4.5,0.08,10) ;MODAL USER CYCLE
N70 G00 X25 Y25
N80 G00 X100 Y100
N90 G00 X175 Y175
N100 MCALL                   ;DESELECT MODAL USER CYCLE
N110 T="SOLID DRILL25"
N120 G95 S850 M03
N130 MCALL MY_CYCLE(1,-12,0.12,10) ;Modal call user cycle

```

Remarques

```

N140 G00 X25 Y25
N150 G00 X100 Y100
N160 G00 X175 Y175
N170 MCALL ;DESELECT MODAL USER CYCLEg
;
N180 G00 Z100 ;RETRACTION
N190 G00 X-100 Y300
200 M05 ;SPINDLE STOP
;
;-----
N210 M30 ;END
;-----

```

```

;=====
;MY_CYCLE
;=====
;
PROC MY_CYCLE (REAL _SAFE, REAL _DEPTH, REAL _FEED, REAL _RETR) DISPLOF SAVE

IF _SAFE <0 GOTOF _ERROR1
ENDIF

IF _DEPTH >0 GOTOF _ERROR2
ENDIF

G00 Z=_SAFE
G01 Z=_DEPTH F=_FEED
G00 Z=_RETR

;-----
RET
;-----

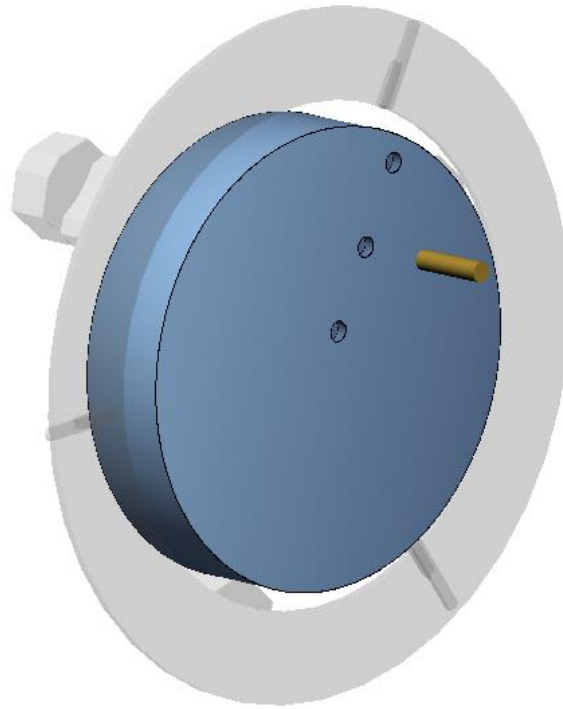
_ERROR1:
MSG("THE SAFETY CLEARANCE MUST BE GREATER THAN ZERO!")
M00
GOTOB _ERROR1

_ERROR2:
MSG("THE DRILLING DEPTH MUST BE LESS THAN ZERO")
M00
GOTOB _ERROR2

```

Remarques

La programmation de la pièce est terminée.



Tâche additionnelle :

Le cycle doit être modifié. Il doit être possible de travailler avec une distance de sécurité négative. Le perçage débute toujours à la surface du matériau.
Exemple : Le perçage commence dans une concavité.

Remarques

Description du module :

Le travail avec un cycle avancé auto-programmé est décrit dans ce module.
Le cycle avancé concerne le programme entier.

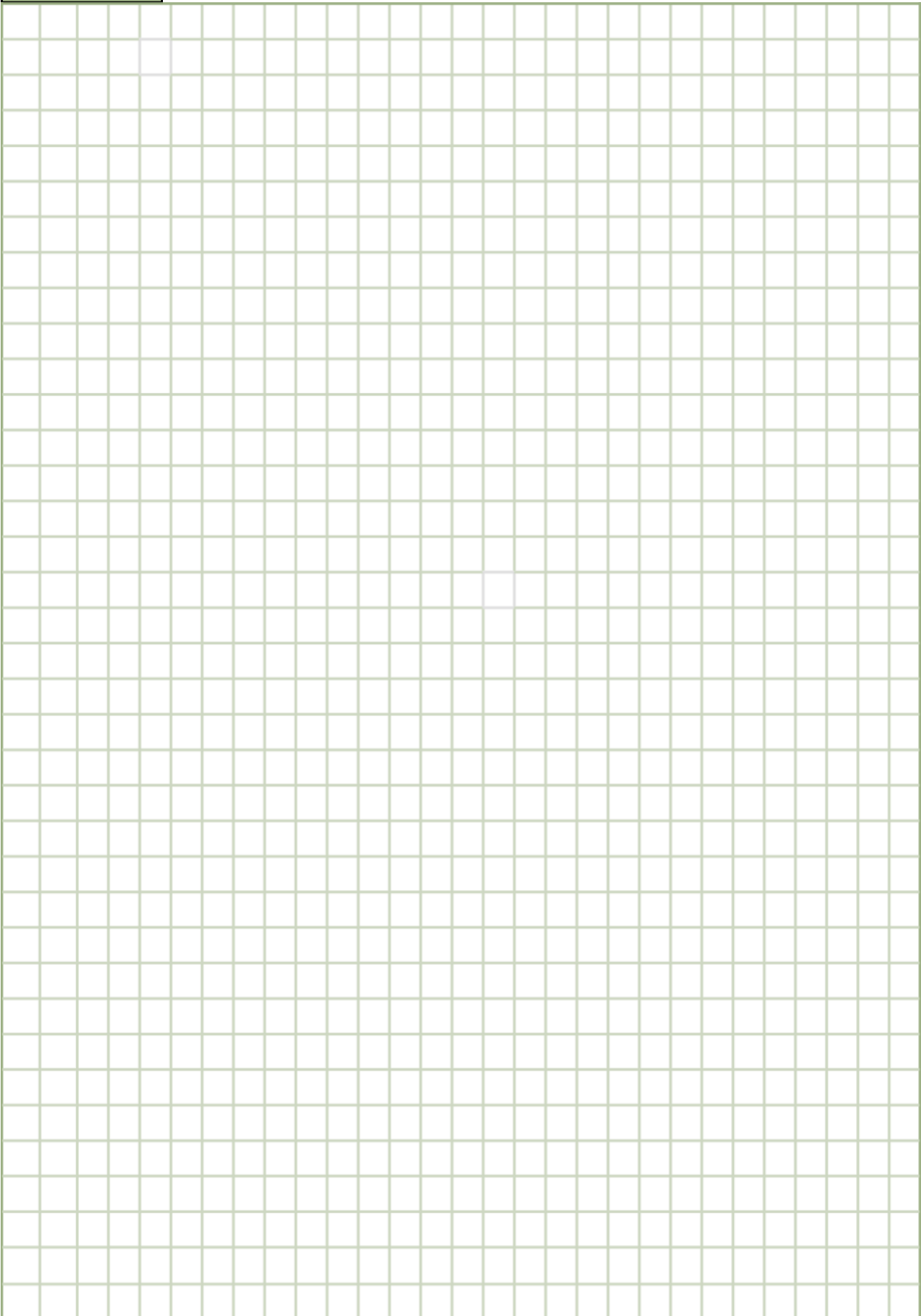
Cas d'utilisation :

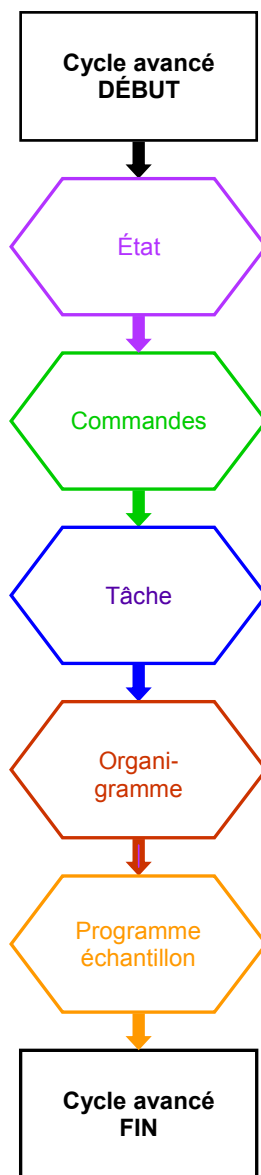
Après avoir complété ce module, vous serez en mesure d'écrire un cycle avancé qui sera effectif dans tous les programmes.

Contenu :

Explication de la structure d'un cycle avancé

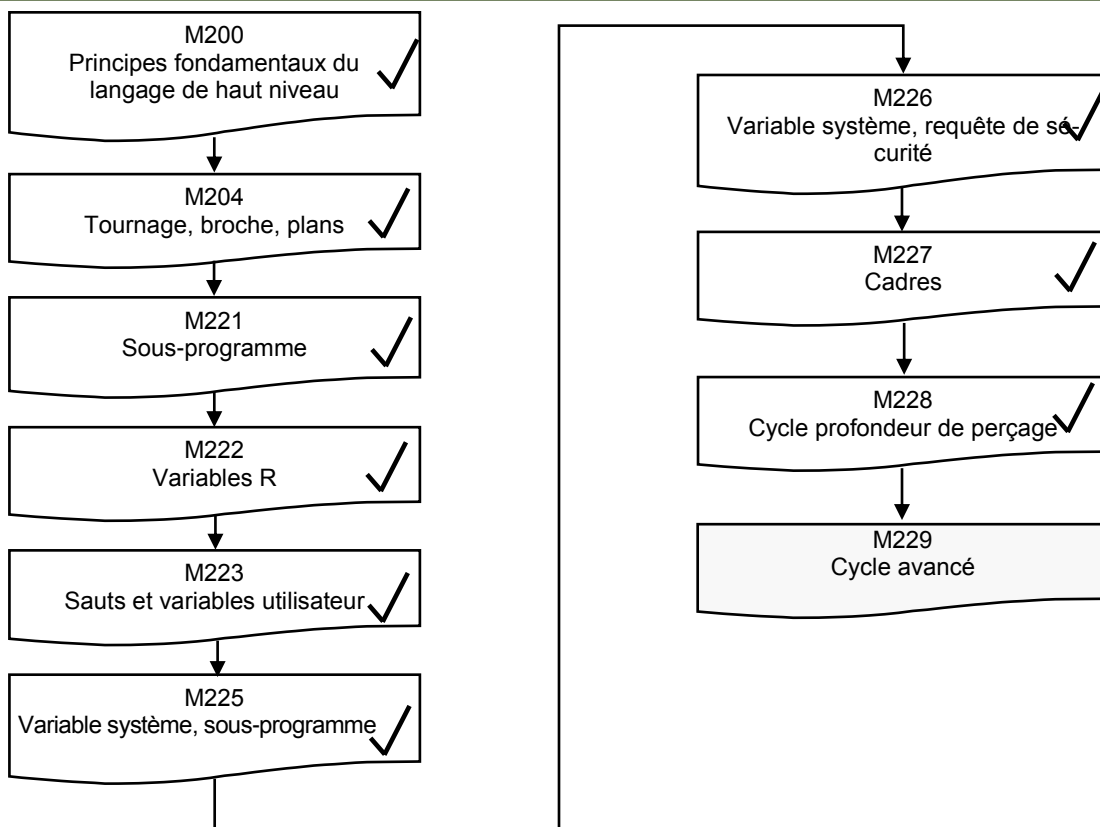
Explication des variables système et des commandes utilisées





Remarques

État :



Commandes :

Commande	Signification
PROC	Première instruction d'un programme
DISPLOF	L'affichage du bloc actuel est supprimé
SAVE	Enregistrement des fonctions G modales
ABS	Dimension absolue
ENDIF/ELSE	Boucle de programmation

Remarques

<u>Commande</u>	<u>Signification</u>
SBLOF	Aucune exécution d'un bloc seul
SBLON	Exécution d'un bloc seul possible
LOOP	Début de la boucle sans fin
ENDLOOP	Ligne de fin d'une boucle sans fin
ENDIF/ELSE	Boucles de programmation
SETAL	Configurer les alarmes

PROC :

Cette commande est utilisée pour les sous-programmes dont les paramètres doivent être remplacés par ceux du programme appelant.

Exemple :

Le cycle avancé est toujours un sous-programme. Le cycle avancé doit être actif dans le programme principal, c'est-à-dire que les valeurs programmées et les paramètres doivent être insérés dans le programme principal.

Si un sous-programme satisfait à cette condition, la première commande sera toujours "PROC".

Tous les cycles d'origine Siemens commencent par cette commande.

DISPLOF :

L'affichage du bloc actuel peut être supprimé avec cette commande.

Exemple : Si les cycles sont appelés dans un programme principal, les blocs NC individuels du cycle avancé actuel ne sont pas visibles. Ils sont supprimés. Les cycles sont généralement programmés comme suit.

SAVE :

Avec la commande SAVE, les fonctions G modales et le cadre actuel actif avant l'appel du sous-programme sont enregistrés par le programme principal et sont de nouveau actifs après le retour.

ABS :

La valeur se réfère toujours à une dimension absolue.

ENDIF/ELSE :

Ces commandes sont utilisées si la boucle de programmation contient un bloc de programme alternatif.

Si la condition IF est vraie, le bloc IF est exécuté.

Si la condition IF est fautive, le bloc de programme après la commande ELSE est exécuté.

La commande ENDIF marque la fin de la boucle IF. Le programme revient au début de la boucle.

SBLOF :

Dans les programmes dans lesquels la fonction SBLOF est programmée, les blocs suivants sont exécutés entièrement, même lorsque l'exécution d'un bloc seul est active.

Exemple :

Une opération d'usinage cohérente ne doit pas être effectuée dans des blocs seuls, car l'outil ou la pièce pourrait être endommagé(e).

Remarques

SBLON :

Après cette commande, les blocs NC suivants sont exécutés en blocs seuls.

LOOP/ENDLOOP :

Une boucle sans fin est généralement programmée avec ces commandes.

Le début de la boucle est programmé avec la commande "LOOP".

ENDLOOP marque la fin de la boucle et induit un retour à "LOOP", le début de la boucle.

Exemple :

Cette commande est utilisée lorsqu'une opération d'usinage particulière est répétée très souvent.

Un compteur peut être intégré dans cette boucle de programmation via des variables R.

Lorsqu'un certain nombre d'opérations d'usinage est atteint, un saut est effectué vers une autre section du programme.

SETAL :

Une alarme peut être programmée avec cette commande.

La fonction SETAL doit être programmée dans un bloc NC séparé, avec le numéro d'alarme.

Les numéros d'alarme et les alarmes sont structurés comme suit :

60000 à 64999, numéros d'alarme réservés pour les cycles Siemens.

65000 à 69999, numéros d'alarme disponibles pour les utilisateurs. Ceux-ci peuvent également bénéficier de textes.

Exemple :

N100 SETAL (65000) ;Set alarm no. 65000.

Utilisation des cycles dans SINUMERIK :

Pour Siemens, les cycles sont toujours des sous-programmes se terminant par ".SPF".

Leur philosophie est la suivante :

Dans la zone d'opération à



il existe trois autres dossiers dans le dossier Cycles.

Cycles			
+	User cycles	DIR	06/29/16 9:17:50 AM
+	Manufacturer cycles	DIR	05/24/16 2:24:51 PM
+	Standard cycles	DIR	07/04/16 12:18:29 PM

Standard cycles

Tous les cycles de SINUMERIK Operate se trouvent dans ce dossier. Ces cycles sont des sous-programmes qui peuvent être ouverts et modifiés. Toutefois, lorsque le contrôle est rechargé, les cycles reviennent à leur état d'origine.

Manufacturer cycles

Les cycles en provenance du fabricant sont placés dans ce dossier.

Les cycles modifiés sont actifs et stockés. Ils ne peuvent être modifiés que par le fabricant de la machine.

User cycles

Les cycles rédigés par l'opérateur sont stockés dans ce dossier.

Les cycles modifiés sont actifs et stockés.

Remarques

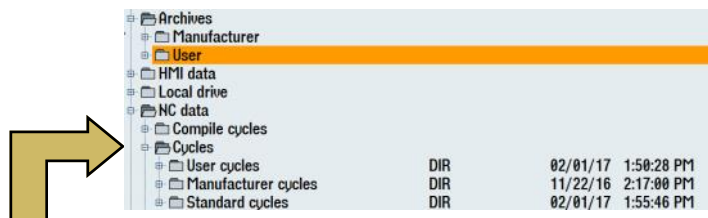
Procédure :

Lorsqu'un cycle utilisateur avancé est créé, il n'a pas besoin d'être entièrement écrit.

Il est possible, par exemple, de modifier les cycles standard.

Si vous souhaitez modifier un cycle avancé existant, procédez comme suit :

Le cycle avancé standard du dossier "Standard cycles" est programmé dans le dossier "User cycles".



Copier le cycle avancé de "Standard cycles" à "User cycles"

Le nom du cycle avancé est conservé, les modifications sont effectuées et le contrôle est redémarré.
Après le chargement, le cycle avancé modifié est disponible.

Remarque : Il n'est pas recommandé de modifier les cycles standard mais, si vous devez le faire, soyez prudent.

La routine suivante s'applique lors de l'appel d'un cycle avancé :

Le système vérifie si le cycle avancé appelé se trouve bien dans le dossier "User cycles". Si c'est le cas, il est exécuté.
Mais si le cycle avancé appelé ne figure pas dans le dossier "User cycles", mais dans le dossier "Manufacturer cycles", il est exécuté.

Si le cycle avancé appelé ne se trouve que dans le dossier "Standard cycles", il est utilisé.

Cette procédure garantit que le tout dernier cycle avancé du programme soit **toujours** utilisé et que, par conséquent, les modifications soient activées.

Si vous écrivez un nouveau cycle avancé, il sera stocké directement dans le dossier "User cycles".

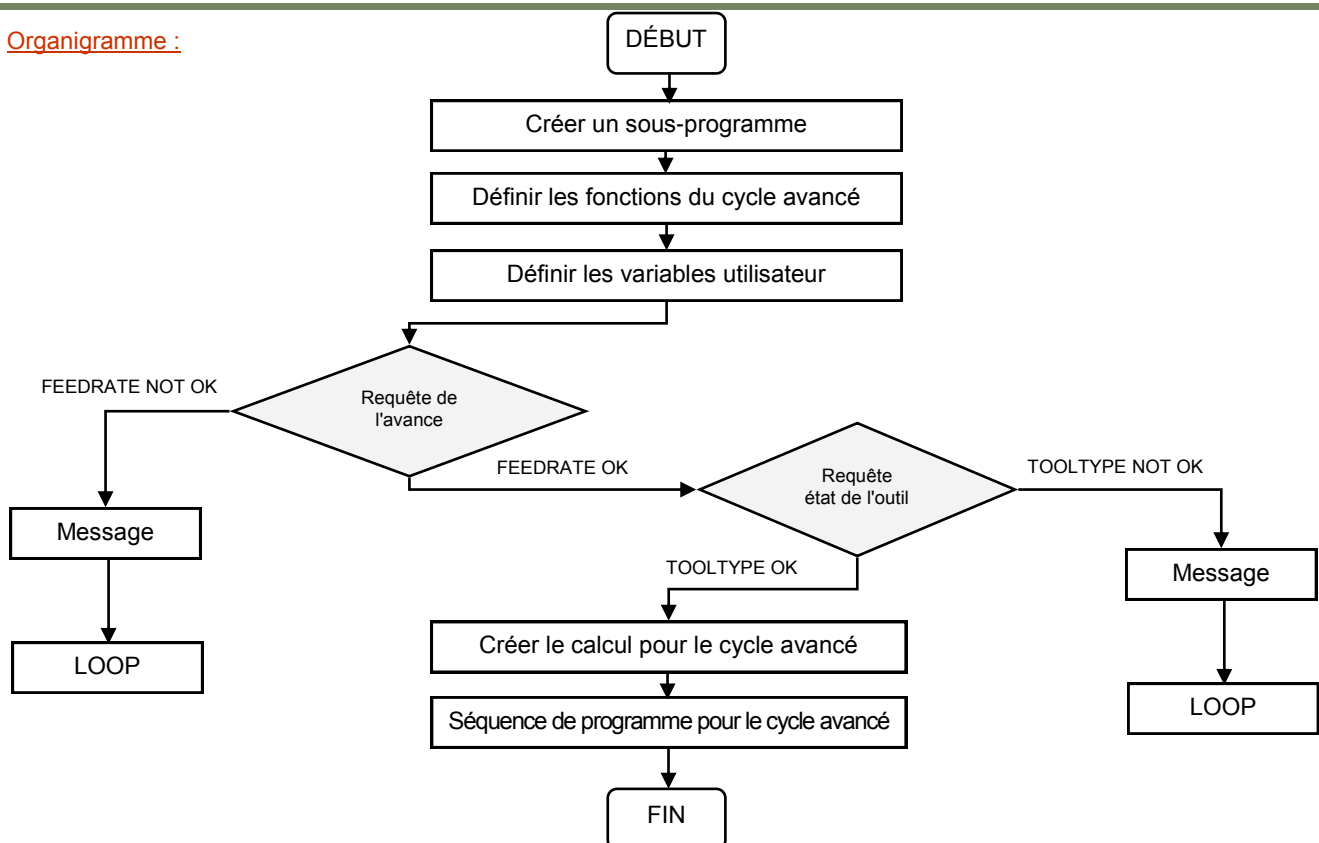
Si un cycle avancé propre est appelé, l'opérateur n'aura aucune influence directe sur lui parce qu'en général, il n'est pas visible et que l'opérateur ne sait pas où le trouver.

Remarques

Tâche :**Description de la tâche**

Un cycle de perçage avancé doit être programmé et utilisé comme cycle utilisateur avancé.

- ⇒ Le cycle avancé doit être programmé dans le programme principal, mais ne doit pas être visible.
- ⇒ Le cycle avancé doit contenir les valeurs suivantes : Position de départ, rétractation, profondeur de perçage complète, profondeur de perçage à la première étape, avance, avance pour la première étape, profondeur d'usinage maximale, profondeur d'usinage minimale.
- ⇒ Si aucune avance n'a été programmée, une boucle de programmation doit être créée, qui ne peut être annulée qu'avec RESET.
- ⇒ Le type d'outil doit être vérifié. Si ce n'est pas correct, une boucle de programmation doit être créée, qui ne pourra être annulée qu'avec RESET.
- ⇒ Avant l'usinage, l'eau de refroidissement doit être mise en marche par l'intermédiaire d'une fonction M.
- ⇒ Le perçage jusqu'à la première profondeur d'usinage doit être réalisé en une seule fois. Pas en mode bloc seul.
- ⇒ La poussée suivante doit être incrémentale jusqu'à ce que la profondeur de perçage soit atteinte à la dernière poussée.
- ⇒ Le trou doit être ensuite enlevé par copeaux.

Organigramme :

Remarques

Programme échantillon :

Cycle avancé

```

PROC DRILL (REAL _STARTP, REAL _RETRAC, REAL _DEPTH_TOTAL, REAL _DEPTH_1, REAL _FEED, REAL
 _FEED_1, REAL _DEPTH_MAX, REAL _DEPTH_MIN)SAVE;DISPLOF
;-----
DEF REAL _DEPTH_CAL
DEF REAL _LAST_STEP
DEF INT _STEP
DEF REAL _STEP_EV
DEF REAL _ACTPOS
DEF INT _TOOLTYPE
;-----

_DEPTH_TOTAL=ABS(_DEPTH_TOTAL)
_DEPTH_1=ABS(_DEPTH_1)

IF _FEED==0 GOTOF _ERROR2
ENDIF

_TOOLTYPE=$TC_DP1[$P_TOOLNO,1]
IF (_TOOLTYPE<200) OR (_TOOLTYPE>299) GOTOF _ERROR1

```

```

;-----

_DEPTH_CAL=( _DEPTH_TOTAL)-(_DEPTH_1)
_STEP=ROUNDUP(_DEPTH_CAL)/(_DEPTH_MAX)
_LAST_STEP=_DEPTH_CAL-( _STEP*_DEPTH_MAX)
IF _LAST_STEP< _DEPTH_MIN
  _LAST_STEP=_DEPTH_MIN
  _DEPTH_CAL=_DEPTH_CAL-_DEPTH_MIN
  _STEP_EV=_DEPTH_CAL/_STEP
ELSE
  _DEPTH_CAL=_DEPTH_MAX*_STEP
  _STEP_EV=_DEPTH_CAL/_STEP
ENDIF
;-----

```

Remarques

```
G0 Z=_RETRAC

M=$MCS_M_CODE_COOLANT_1_ON

G1 Z=_STARTP F=_FEED_1
G1 Z=-_DEPTH_1
G0 Z=_RETRAC
G0 Z=(-_DEPTH_1)+1
SBLOF
G1 Z=(-_DEPTH_1) F=_FEED
LOOP
G1 Z=IC(-_STEP_EV)
SBLON
_ACTPOS=$AA_IW[Z]
G0 Z=_RETRAC
IF _ACTPOS==( _DEPTH_TOTAL-_LAST_STEP)*(-1) GOTOF _END
G0 Z=_ACTPOS+1
SBLOF
G1 Z=IC(-1)
ENDLOOP
_END:
```

```
;-----
G0 Z=_ACTPOS+1
G1 Z=_DEPTH_TOTAL*(-1)
G0 Z=_RETRAC

;-----
M17

_ERROR1:
MSG("INCORRECT TOOL TYPE")
M0
GOTOB _ERROR1

_ERROR2:
SETAL(61003)
M0
GOTOB _ERROR2

M17
```

Remarques