

IUT d'EVRY VAL D'ESSONNE

YE Carlos

MOCHET Matthieu

TOURAIS Aubain

CHENEVIÈRE Ethan

PROJET FANUC OLYMPIADE 2023-2024

Sertisseuse

Table des matières

Table des matières

1) Mission 1 : Calcul du moteur en poste amont Z	3
Partie calcul de vitesse	3
A partir du calcul de l'accélération	4
2) Mission 2 : Création des pages Opérateurs, Etat Machine, Maintenance et Gestion des Alarmes	5
Page opérateur	6
État machine	11
Page maintenance.....	12
Gestion Alarmes.....	14
3) Mission 3 : Synoptique de la machine	17
4) Mission 4 : Le Programme ISO	18
5) Mission 5 :FANUC LADDER.....	22
Page simulations des capteurs :.....	23
Page mode automatique :	26
Page code M :	30

1) Mission 1 : Calcul du moteur en poste amont Z

A partir des informations ci-dessous :

Données techniques :

- Système pignon / crémaillère (pignon diamètre 60 mm) axe horizontal
- Masse totale en mouvement : 500 kg (250 kg sur la poutre 1 et 250 kg sur la poutre 2)
- Vitesse maxi = 0,35 m/s, Accélération maxi = 0.35 m/s²
- Aucun effort de sertissage à prévoir
- Réducteur planétaire
- Réduction de 20

On est censé pouvoir calculer un élément discriminant qui va pouvoir nous permettre de choisir le moteur situé dans le document ci-dessous:

Documents :

- Description_Moteur_Alpha_Serie
- Parametre_Servo_B-65270EN_10_04

Voici ci-dessous notre piste de raisonnement :

Partie calcul de vitesse

Vitesse maxi en sortie de réducteur -> $V_{max_red} = 0.35 \text{ m/s}$

Vitesse maxi avant le réducteur de 20 -> $V_{max_Nred} = V_{max_red} * 20 = 7 \text{ m/s}$

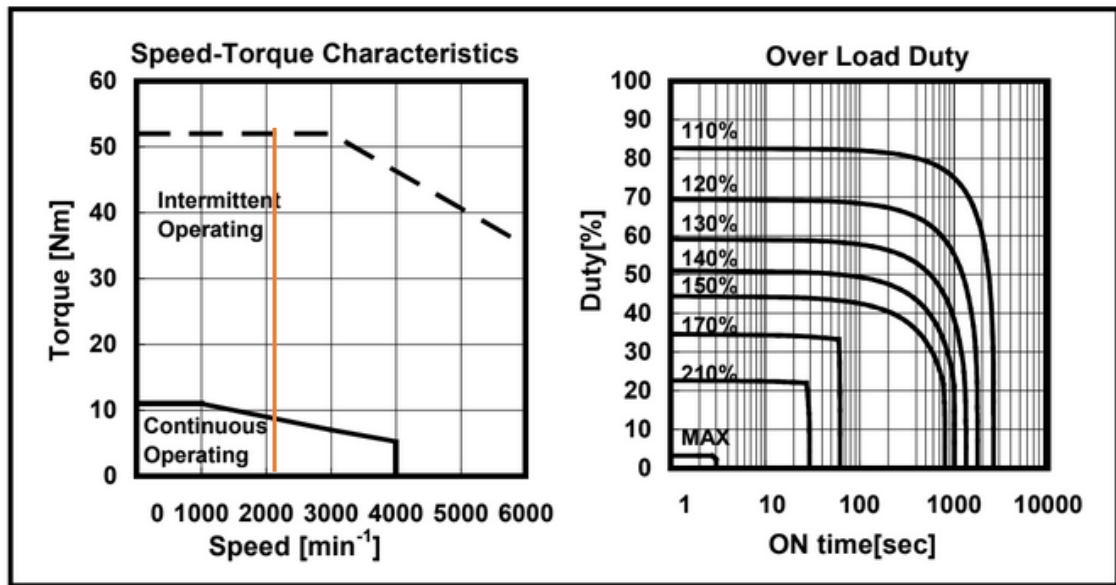
Vitesse angulaire maxi avant le réducteur de 20 :

$$V_{max_ang_Nred} = \left(\frac{V_{max_Nred}}{r} \right) * \left(\frac{60}{2\pi} \right) = 2228 \text{ tr/min}$$

Sur la doc technique, on peut obtenir le Torque_Max = $C_{max} = f(V_{max_ang_Nred})$

Model α iS 12/6000 HV-B

Specification: A06B-2237-B□0□



Pour $V_{max_ang_Nred} = 2228 \text{ tr/min} \Rightarrow C_{max} = 52 \text{ Nm}$

On peut vérifier ce résultat :

Maximum Torque (*)	T_{max}	52 Nm 531 kgfcm
--------------------	-----------	--------------------

A partir du calcul de l'accélération

Accélération en sortie de réducteur $\rightarrow a_{red} = 0.35 \text{ m/s}^2$

Accélération avant réducteur $\rightarrow a_{Nred} = 7 \text{ m/s}^2$

D'après la doc technique

$J_m = 0.00228 \text{ kg.m}^2$ (donnée par la documentation technique)

Moment of Inertia of Rotor	J_m	0.00228 kgm ² 0.0233 kgfcm ²
----------------------------	-------	---

$C_{friction} = 0.4 \text{ Nm}$ (donnée par la documentation technique)

Static friction	T_f	0.4 Nm 4 kgfcm
-----------------	-------	-------------------

$$J_m \cdot dw/dt = 0.00228 \cdot 2228 = 5.08 \text{ Nm}$$

$$C_{max} = m \cdot a_{Nred} \cdot r = 250 \cdot 7 \cdot 0.03 = 52.5 \text{ Nm}$$

Si on inclut le moment d'inertie et le couple de friction, on obtient

$$C_{max} - J_m \cdot dw/dt - C_{friction} = 47.02 \text{ Nm}$$

Ce qui correspond au dernier moteur de la documentation technique : **Model α iS 12/6000 HV-B**

2) Mission 2 : Création des pages Opérateurs, Etat Machine, Maintenance et Gestion des Alarmes

Dans cette partie, nous allons mettre en place l'interface de communication entre la machine et l'utilisateur.

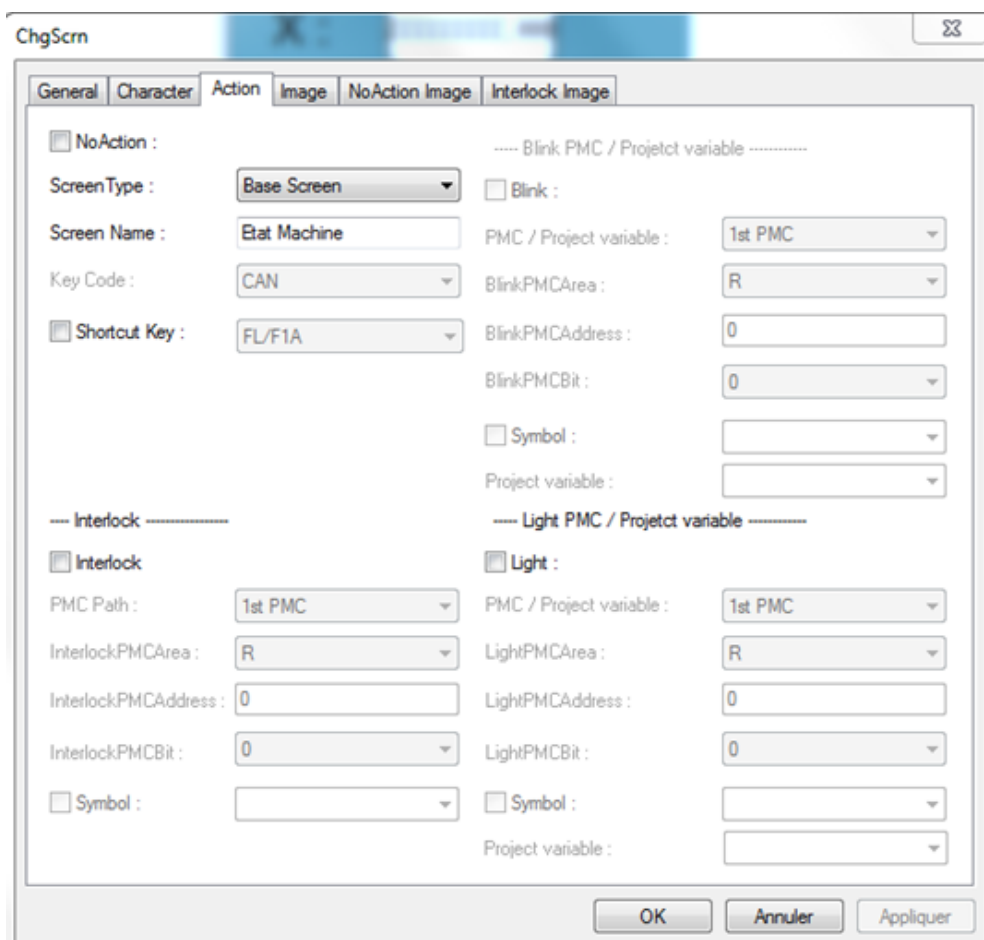
Pour cela, nous utilisons ce que nous appelons une IHM, pour réaliser cette étape, nous utilisons une application qui se nomme « Fanuc Picture ».

Afin de répondre au cahier des charges nous créons un total de quatre fenêtres.

- Page Opérateur
- Page Etat Machine
- Page Maintenance
- Page Gestion des Alarmes

Chacune de ces pages auront des spécificités à elles.


De plus, nous pourrons changer de page via les boutons se trouvant en bas des pages avec la configuration suivante pour chaque bouton :



The screenshot shows the 'ChgScrn' (Change Screen) dialog box. It has several tabs: General, Character, Action, Image, NoAction Image, and Interlock Image. The 'General' tab is selected. Inside, there are two main sections: 'NoAction' and 'Interlock'.
 In the 'NoAction' section:
 - 'Screen Type' is set to 'Base Screen'.
 - 'Screen Name' is 'Etat Machine'.
 - 'Key Code' is 'CAN'.
 - 'Shortcut Key' is 'FL/F1A'.
 - 'Blink PMC / Project variable' is set to '1st PMC'.
 - 'BlinkPMCArea' is 'R'.
 - 'BlinkPMCAddress' is '0'.
 - 'BlinkPMCBit' is '0'.
 - 'Symbol' and 'Project variable' fields are empty.
 In the 'Interlock' section:
 - 'Interlock' checkbox is checked.
 - 'PMC Path' is '1st PMC'.
 - 'InterlockPMCArea' is 'R'.
 - 'InterlockPMCAddress' is '0'.
 - 'InterlockPMCBit' is '0'.
 - 'Symbol' and 'Project variable' fields are empty.
 At the bottom, there are three buttons: 'OK', 'Annuler', and 'Appliquer'.

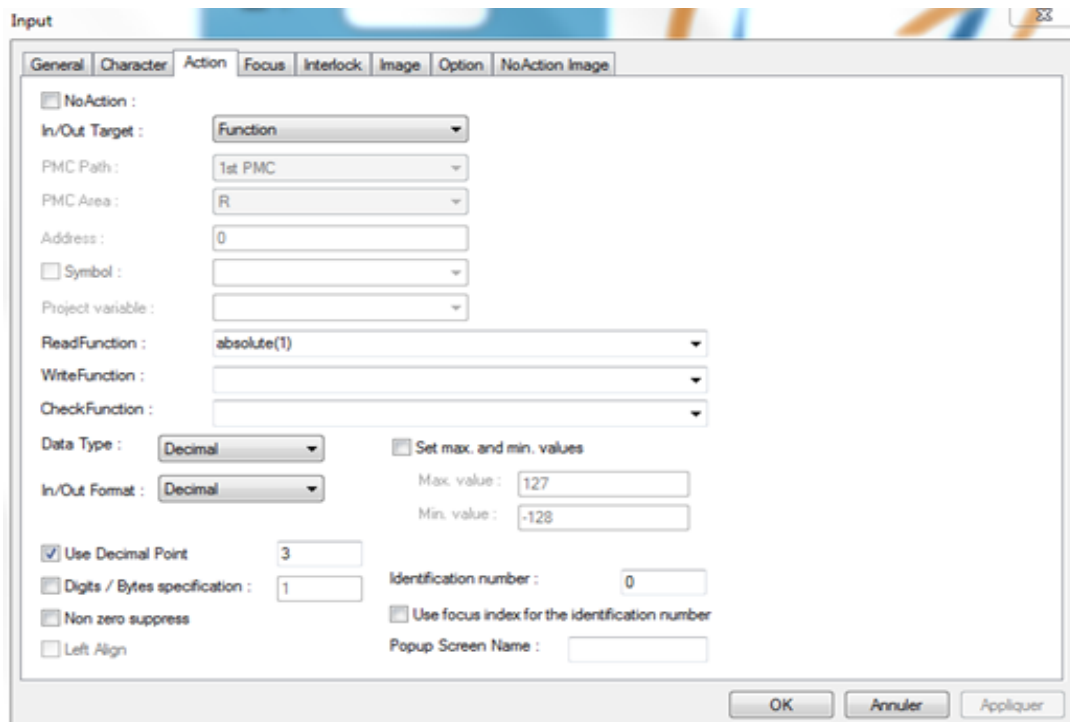
« **Screen Name** » correspondant au nom de la page souhaitée.

Page opérateur



Cette page aura pour objectif de permettre à l'utilisateur de pouvoir :

- Lire le positionnement des axes X/U – Y/V et Z sur le côté gauche de la fenêtre, ces valeurs seront récupérées grâce à une fonction qui se nomme : **absolute()** via l'objet « **Input** » de Fanuc Picture (Cette fonction permet de récupérer la valeur des axes en absolue). Le paramètre à l'intérieur se décide en fonction de l'axe à lire (1 correspondant à X, 2 à Y et 3 à Z).



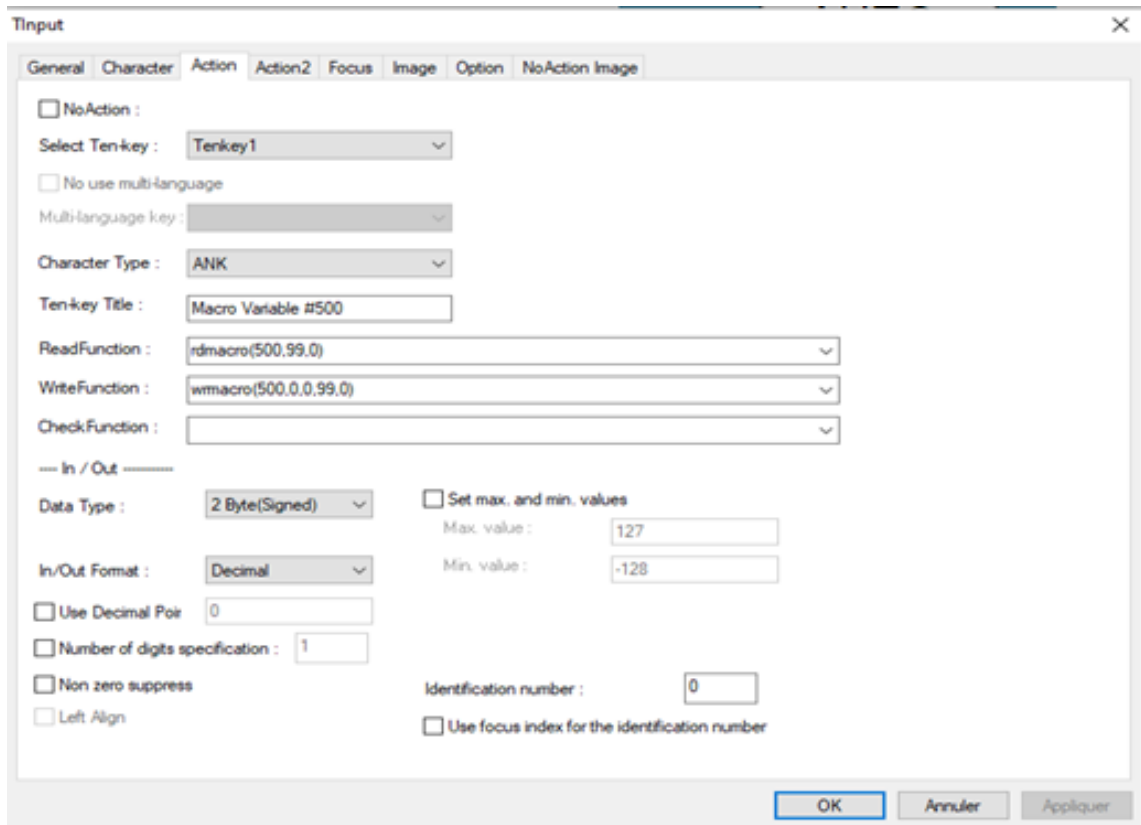
Néanmoins, pour les axes U et V, n'étant pas présents sur la sertisseuse à 3 axes, ne sont pas programmés, afin de pouvoir les simuler, nous effectuons la synchronisation des axes (Axe U avec l'axe X, puis l'axe V avec l'axe Y)

- Modifier les valeurs de déplacement des axes X, Y et Z grâce à une fonction qui se nomme : **rdmacro()** via l'objet « Ten-key » de Fanuc Picture.

Voici comment faut-il paramétrer la fonction **rdmacro**, le 500 correspond ici au numéro de la variable sauvegardée #500 de l'axe X, variable qu'on utilise dans le code ISO pour sauvegarder la position de l'axe correspondant. Le 99 correspond au type et le 0 au mode.

Afin de permettre l'écriture via le code ISO, nous appliquons une autre fonction : **wrmacro()**.

Voici un exemple de la configuration pour l'axe X (cette configuration est identique pour l'axe Y et Z, mais nous changeons uniquement la valeur des variables qui leur correspondent, Y avec #503 et Z avec #506)



- Visualiser et sélectionner les modes de fonctionnement (Auto/Edit/MD/Manuel)

Ces boutons sont directement présents sur le clavier mais nous pouvons appuyer sur ces boutons directement sur le Fanuc picture pour activer ces différents modes :

- Auto permet d'exécuter le programme ISO et Ladder.
- Edit permet d'éditer et sélectionner le code ISO via la CNC.
- MDI (non programmé) permet de lancer une ligne de code avec le bouton « Start cycle »

- Manuel permet de faire bouger directement les axes X, Y et Z avec la télécommande, ce qui revient à faire la fonction « Handle » du clavier.

- Sélectionner le Départ ou l'Arrêt du cycle via des boutons.

Ces boutons permettent de mettre en marche ou d'arrêter le programme (cycle start et cycle stop du clavier)

- La partie alarmes et messages sera mise dans la page 4 : Gestion des alarmes, étant donné que sur cette page nous pouvions faire l'objet de mettre ce que nous souhaitions, nous avons ainsi décidé de réunir tous les messages et alarmes sur cette page.

Pour les boutons des Modes de fonctionnement ou pour le choix du Départ et de l'Arrêt du cycle, voici comment nous nous y sommes pris :

Une fois que nous appuyons sur ces boutons, la LED d'affichage qui était en rouge s'allume et change de couleur en devenant vert, ce qui permet de dire sous quel mode nous sommes.

A noter, que pour ne pas avoir le problème d'être sous plusieurs mode, nous appliquons ce que nous appelons le « Switch Type » sous « Alternate », ce qui nous permet que si nous avons plusieurs modes dans le même groupe, en alternate, lorsque l'on aura un bouton enclenché, cela éteindra tous les autres automatiquement.

Les modes sont sous le groupe 2, et les boutons Départ/Arrêt sont sous le groupe 10.

Nous avons pour chaque bouton, le « Light PMC » qui permet d'allumer ou non sa LED via un Y récupérer sous la PMC qui est identique entre le bouton et la LED, et afin d'enclencher ou non le bouton, nous utilisons un R qui est la condition d'activation dans la PMC.

Pour les boutons, voici la configuration :

Op8tn

General Character Action Image NoAction Image Interlock Image

☐ NoAction :
☐ Keep State :
 SwitchType : Alternate
 GroupNo : 2
☐ Shortcut Key : FL/F1A

----- Blink PMC / Project variable -----
☐ Blink :
 PMC / Project variable : 1st PMC
 BlinkPMCArea : R
 BlinkPMCAAddress : 0
 BlinkPMCBit : 0
☐ Symbol :
 Project variable :

----- Interlock 1 -----
 Interlock1 : No Use Interlock
 PMC Path : 1st PMC
 InterlockPMCArea1 : R
 InterlockPMCAAddress1 : 0
 InterlockPMCBit1 : 0
☐ Symbol :

----- Interlock 2 -----
 Interlock2 : No Use Interlock
 PMC Path : 1st PMC
 InterlockPMCArea2 : R
 InterlockPMCAAddress2 : 0
 InterlockPMCBit2 : 0
☐ Symbol :

----- Light PMC / Project variable -----
☐ Symbol :
 Project variable :

----- OUT PMC / Project variable -----
 PMC / Project variable : 1st PMC
 OutPMCArea : R
 OutPMCAAddress : 7502
 OutPMCBit : 7
☐ Symbol :
 Project variable :

OK Annuler Appliquer

Pour les LEDs, voici la configuration :

Lamp

General Character **Action** Image Option NoAction Image

☐ NoAction :

PMC / Project variable : 1st PMC

Watch PMC Area : Y

Watch Address : 102

Watch Bit : 7

☐ Symbol :

Project variable :

☐ UseBlink :

PMC / Project variable : 1st PMC

BlinkSignalArea : R

BlinkSignalAddress : 0

BlinkSignalBit : 0

☐ Symbol :

Project variable :

OK Annuler Appliquer

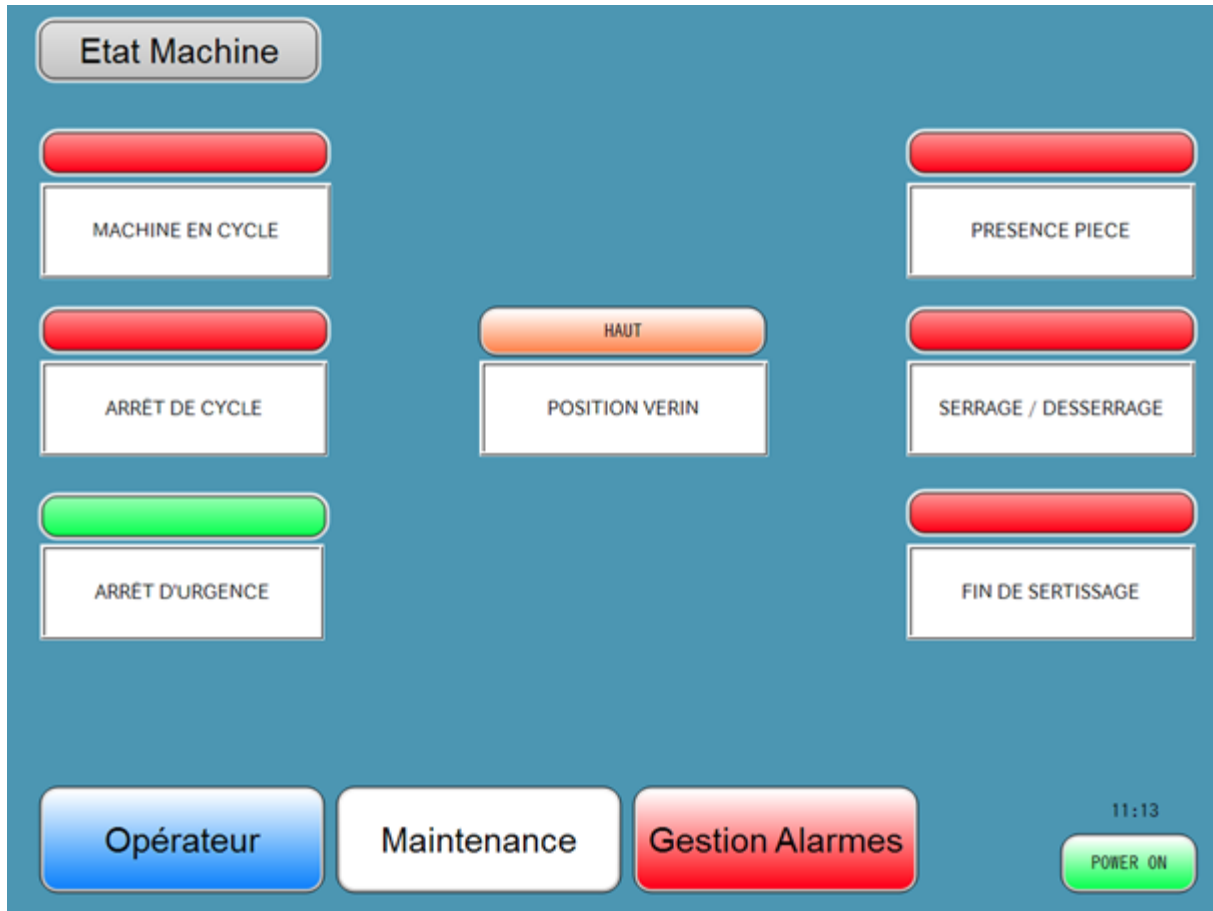
BONUS :

- Nous avons rajouté une LED qui peut switcher entre le mode « power on » et « power off », selon si la machine est allumée ou éteinte.
- L'utilisateur pour piloter les vérins de Serrage/Cadrage et les mâchoires de sertissages via la même option que le choix des valeurs des axes X-Y-Z sous « Ten-Key ».

A partir de cette page, l'utilisateur pourra aller sur les autres pages via les boutons se trouvant en bas. Il pourra en cliquant sur le bouton :

- « Etat Machine » aller directement sur la page « État Machine »
- « Maintenance » aller directement sur la page « Maintenance »
- « Gestion Alarmes » aller directement sur la page « Gestion Alarmes »

État machine



Sur cette page, nous aurons certains états de la machine comme :

- Machine en Cycle
- Arrêt de Cycle
- Arrêt d'Urgence
- Présence Pièce
- Serrage/Desserrage
- Fin de sertissage
- Position Vérin

Pour les LEDs se trouvant à gauche de notre fenêtre, celles-ci s'allumeront et s'éteindront automatiquement.

- Si la machine est en « Cycle Start », nous aurons la LED de la machine en cycle qui s'allumera.
- Si la machine est en « Cycle Stop », nous aurons la LED d'Arrêt de cycle qui s'allumera.
- La LED d'Arrêt d'urgence sera allumé uniquement si nous enfonçons le bouton correspondant, la machine arrêtera son fonctionnement, et la LED sera en vert.

Pour les LEDs se trouvant à droite de notre fenêtre, celles-ci s'allumeront et s'éteindront via des capteurs que nous simulons.

- La LED correspondant à la Présence Pièce sera en vert quand nous simulons la présence d'une pièce au départ, ce qui correspond aussi à la LED que nous retrouverons dans la page 3 : Etat Machine avec « Pièce au poste de : CHARGEMENT »
- La LED Correspondant au Serrage/Desserrage sera activée quand nous appuierons sur le bouton correspondant à sa simulation. (Transition entre Etape 4 et Étape 5 du grafctet)
- La LED Correspondant au Fin de sertissage sera activée quand nous appuierons sur le bouton correspondant à sa simulation. (Transition entre Etape 5 et retour à l'Étape 0 du grafctet)

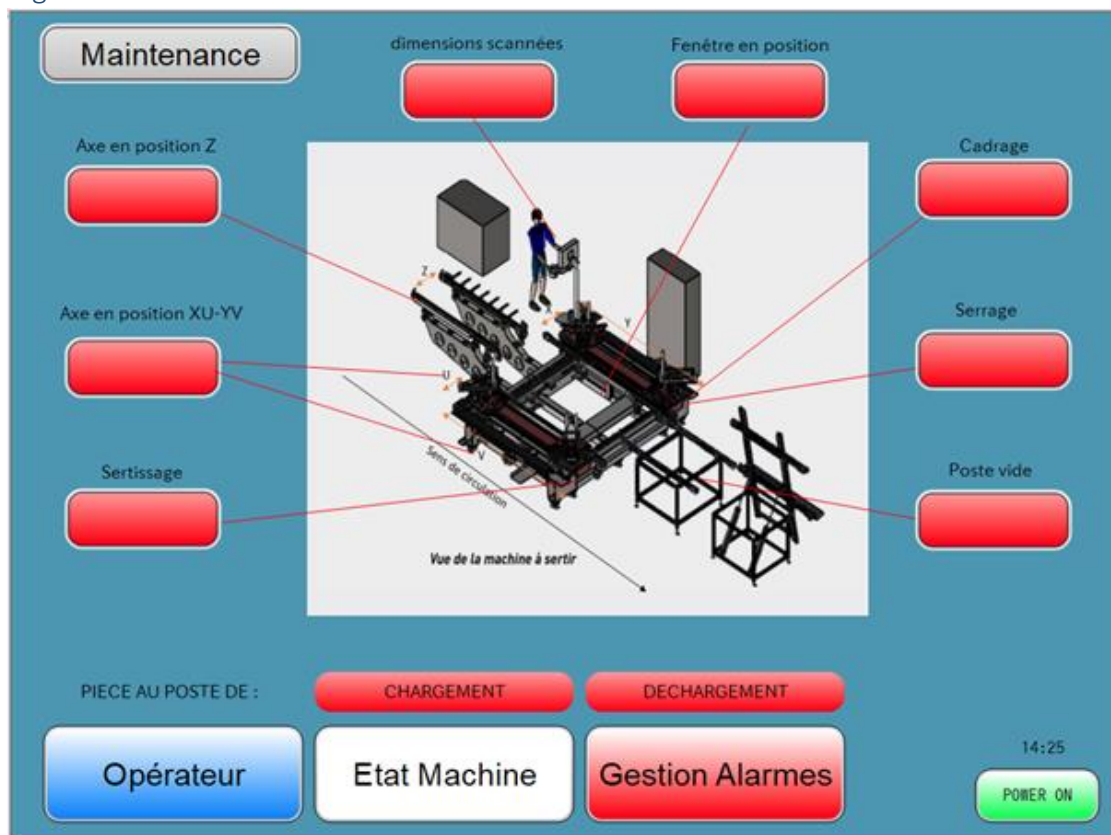
Nous avons aussi la LED centrale, qui permet de savoir si la position de nos vérins est à l'état haut ou bas.

Pour cela, nous avons fait en sorte que si l'utilisateur entre une valeur supérieure à 0 dans le Ten-key du pilotage des vérins, alors l'état des vérins est « haut » mais si l'utilisateur entre comme valeur : 0, alors l'état des vérins est « bas ».

A partir de cette page, l'utilisateur pourra aller sur les autres pages via les boutons se trouvant en bas. Il pourra en cliquant le bouton :

- « Opérateur » aller directement sur la page « Opérateur »
- « Maintenance » aller directement sur la page « Maintenance »
- « Gestion Alarmes » aller directement sur la page « Gestion Alarmes »

Page maintenance



On a dans cette page, des lampes qui vont s'allumer en fonction de l'état du capteur qui sera activé afin de simuler le grafctet.

Nous avons de plus, la synoptique de notre machine se trouvant au centre de la fenêtre avec la liaison des capteurs se trouvant sur la machine en question.

- Les LEDs « Chargement » et « Déchargement » sont des lampes qui vont être activées en fonction de l'état des capteurs simulés par 2 boutons du clavier.

Ils permettent d'indiquer si nous sommes dans l'étape de chargement ou de déchargement de la machine, nous indiquerons par la suite, le message de si nous sommes dans la phase chargement ou dans la phase de déchargement dans la page 4 : Gestion des Alarmes.

Les autres capteurs seront activés si nous enclenchons les boutons qui leurs servent de simulations.

Les capteurs se trouvant autour de notre synoptique sont les transitions de notre grafset.

- Dimensions Scannées : Transition de l'étape 0 à l'étape 1
- Axe en position Z : Transition de l'étape 1 à l'étape 2
- Axe en position XU-YV : Transition de l'étape 2 à l'étape 3
- Cadrage : Transition de l'étape 3 à l'étape 4
- Serrage et Fenêtre en position : Transition de l'étape 4 à l'étape 5
- Sertissage : Transition de l'étape 5 à l'étape 6
- Poste Vide : Transition de l'étape 6 à l'étape 0

A partir de cette page, l'utilisateur pourra aller sur les autres pages via les boutons se trouvant en bas. Il pourra en cliquant le bouton :

- « Opérateur » aller directement sur la page « Opérateur »
- « État Machine » aller directement sur la page « État Machine »
- « Gestion Alarmes » aller directement sur la page « Gestion Alarmes »

Gestion Alarmes



Un message d'alarme va s'afficher dans la partie « ALARMES » une fois que la machine est dans un état qui l'empêche de continuer de fonctionner.

Par exemple :

- Bouton Arrêt d'Urgence enfoncé
- Limite des axes atteints

Etc.

Nous afficherons ici tous les messages qui renvoies un #3000.

Puis nous avons un message « informatif » qui va s'afficher dans la partie « MESSAGES » une fois que la machine reçoit une information qui permet d'informer l'utilisateur d'un état.

Par exemple :

- Pièce en position de chargement
- Pièce en position de déchargement

Etc.

Nous afficherons ici tous les messages qui renvoies un #3006.

Voici ci-dessous comment on programme l'affichage des alarmes :

NDisp

General Character Action Image Option NoAction Image

☐ NoAction :

Action Type : Function

--- Display ---

Data Type : Rectangle String

Format : Decimal

--- PMC ---

PMC Path : 1st PMC

Watch PMC Area : R

Watch Address : 0

☐ Symbol :

--- Function ---

Function Name : ☒ Display :

rdalminfo(1,99,10,99,0)

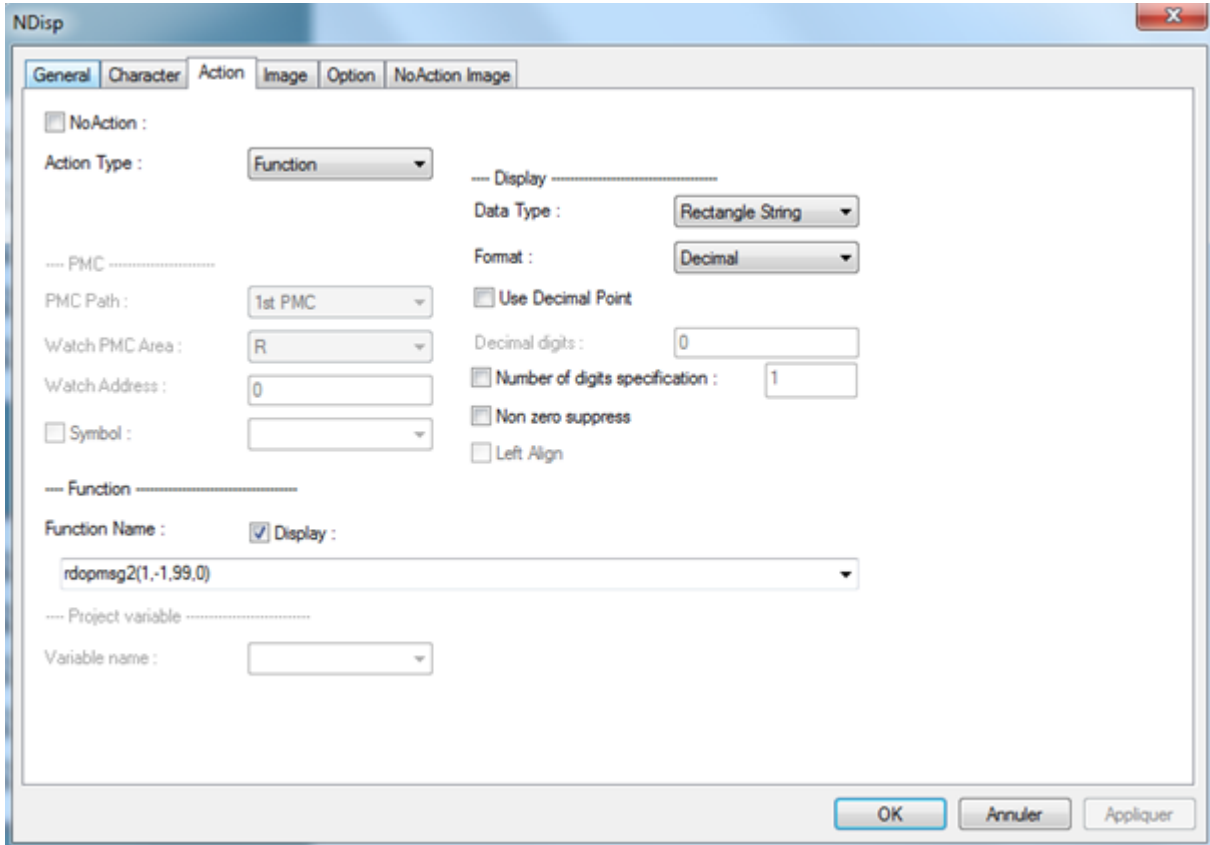
--- Project variable ---

Variable name :

OK Annuler Appliquer

Nous pouvons lire les alarmes via la fonction: « ***rdalminfo()*** » sont trouvables dans la doc technique.

Puis pour les messages, voici la configuration :



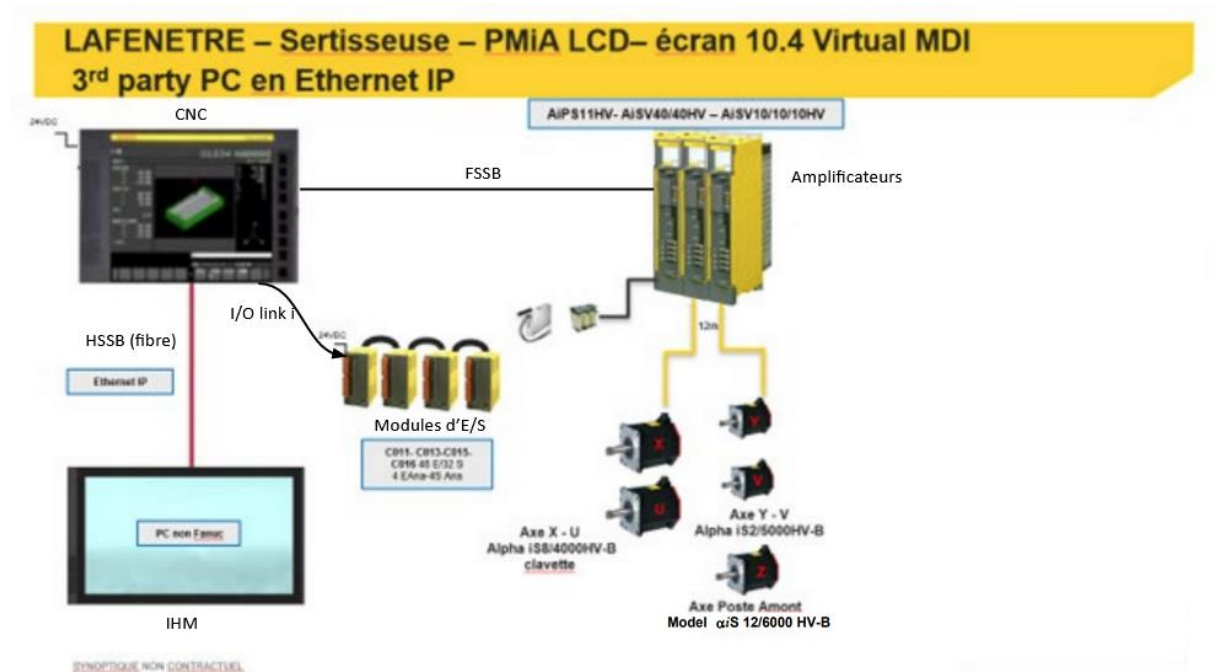
Nous pouvons lire les messages via la fonction: « **rdopmsg2()** » sont trouvables dans la doc technique.

A partir de cette page, l'utilisateur pourra aller sur les autres pages via les boutons se trouvant en bas. Il pourra en cliquant le bouton :

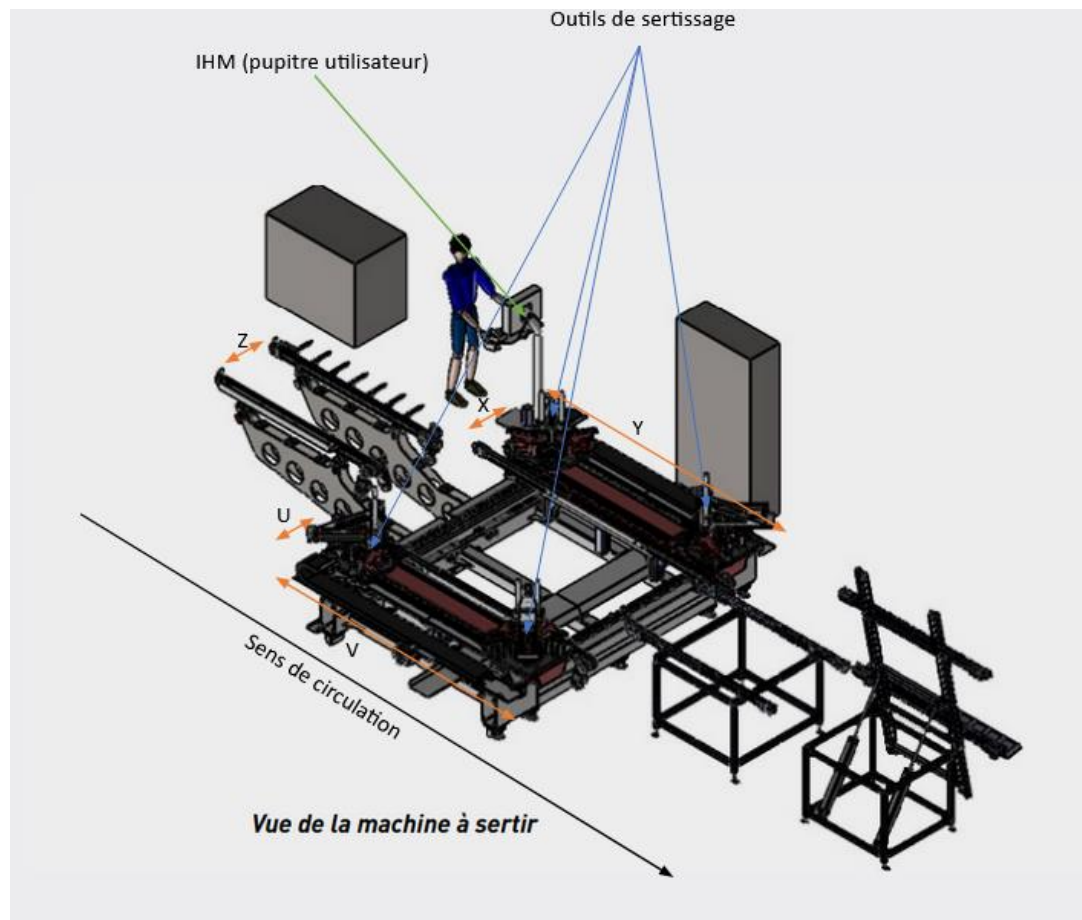
- « Opérateur » aller directement sur la page « Opérateur »
- « État Machine » aller directement sur la page « État Machine »
- « Maintenance » aller directement sur la page « Maintenance »

3) Mission 3 : Synoptique de la machine

Voici ci-dessous, une explication des différents composants de la machine

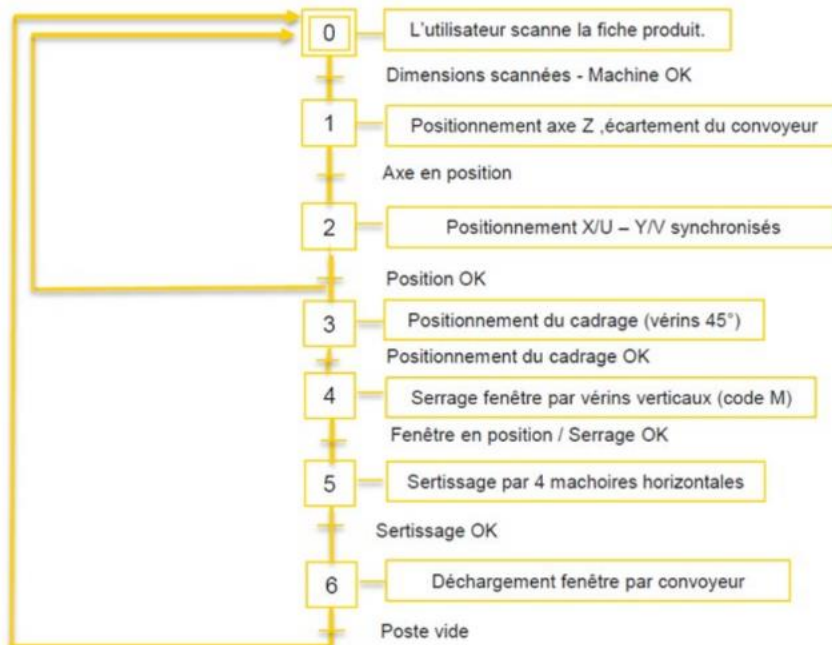


Voici ci-dessous, une explication des différentes étapes de sertissage de la machine



4) Mission 4 : Le Programme ISO

Dans cette partie, nous allons mettre en place le programme permettant le fonctionnement du mode AUTO décrit par le grafcet suivant:



Nous avons identifié dans le grafcet que le rebouclage après "Position OK" était un OU.

```
G54 G90 G94;  
N500;
```

```
; (TEST MESSAGE ERREUR COOR. X-Y-Z)  
IF[#500 LE # 501] OR [#500 GE # 502] OR [#503 LE # 504] OR [#503 GE # 505]  
OR [#506 LE # 507] OR [#506 GE # 508]] THEN #3000 = 12 (ATTENTION LIMITES AXES DEPASSEES);  
IF[#1016 EQ 1] THEN #3006 = 30 (PIECE CHARGEMENT);  
IF[#1017 EQ 1] THEN #3006 = 30 (PIECE DECHARGEMENT);
```

Dans le code un #800 est utilisé, l'initialiser à 0 s'il y a un problème lors du lancement.

G54: Décalage d'origine de pièce

G90: cotation absolue

G94: Avance par minute

Dans les IF, les #3006=30 permet d'envoyer un message qui n'arrête pas le programme

(Le IF est mis sur 2 lignes pour le rendre plus lisible sur le compte rendu). Le but de cette condition IF est d'éviter les surcourses, nous avons décidé d'imposer une marge de 10 pour être sûr que la machine ne s'arrête pas inutilement à cause d'une mauvaise valeur entrée. Les valeurs liées aux surcourses sont stockées dans les variables suivantes:

#501: X:-80 #502: X+:600 #504: Y:-310 #505: Y+:5 #507: Z:-8 #508: Z+:100

Le (THEN #3000=12) avec le message entre parenthèses après permet de déclencher une alarme indiquant que les axes vont être en surcourses.

```
.
; (#Av0 - SCAN)
IF[#1001 EQ 1] AND [#800 EQ 0]] GOTO 999;
```

```
(SCAN)
N999 #1101 = 1;
N999 #800 = 1;
GOTO 10;
```

Dans la partie IF, on vérifie si on a bien reçu le G de la fonction précédente (#1001 EQ 1) et le (#800 EQ 0) permet de savoir dans quelle étape on est et d'être sûr de ne pas entrer à nouveau dans une fonction que l'on aurait déjà faite.

On set le compte rendu à 1 (#1101 =1), on passe le #800 à 1 pour dire qu'on a fait l'étape.

```
; (#AV1 - POSITIONNEMENT AXE Z)
N10 IF[#1002 EQ 1] AND [#800 EQ 1]] GOTO 998;
```

```
(Axe Z)
N998 #1101 = 0;
N998 G01 Z#506 F1000;
N998 #800 = 2;
N998 #1102 = 1;
GOTO 20;
```

Dans la partie IF, on vérifie si on a bien reçu le G de la fonction précédente (#1002 EQ 1) et le (#800 EQ 1) permet de savoir dans quelle étape on est et d'être sûr de ne pas entrer à nouveau dans une fonction que l'on aurait déjà faite.

Dans la partie (axe Z) on remet le compte rendu (F) correspondant à la partie précédente à 0 (#1101=0), on déplace l'axe Z à la position voulue, on utilise la variable #506 pour sauvegarder la valeur de Z, on change la valeur du #800 pour valider l'étape et on passe le F correspondant à 1 (#1102=1). Enfin on utilise un GOTO pour aller à la condition/étape suivante.

```
; (#Av2 - POSITIONNEMENT AXE X-Y)
N20 IF[#1003 EQ 1] AND [#800 EQ 2]] GOTO 997;
```

```
(Axes X/U_Y/V)
N997 #1102 = 0;
N997 G01 X#500 Y#503 F1000;
N997 #800 = 3;
N997 #1103 = 1;
GOTO 30;
```

Dans le IF on fait pareil que dans le précédent.

Dans la partie (axe X/U_Y/V) on fait pareil qu'avec la partie axe Z mais avec les 2 axes X et Y. On utilise les variables #500 et #503 pour sauvegarder les valeurs de X et Y respectivement.

```
; (#AV3 - CADRAGE)
N30 IF[#1004 EQ 1] AND [#800 EQ 3]] GOTO 996;
```

```
.
```

```
(CADRAGE)
N996 #1103 = 0;
N996 #800 = 4;
N996 #1104 = 1;
GOTO 40;
```

Dans le IF on fait pareil que dans le précédent.

Dans la fonction, on reset le compte rendu de la partie précédente (#1103 =0), on passe le #800 à 4 pour valider l'étape et ne pas retourner dedans et on passe le compte rendu de cette partie à 1 (#1104 =1).

```
; (#Av4 - SERRAGE)
N40 IF[#1005 EQ 1] AND [#600 EQ 45] AND [#700 EQ 1] AND [#800 EQ 4]] GOTO 995;

(SERRAGE)
N995 #1104 = 0;
N995 M10;
N995 #700 = 0; (Variable If_Code_M)
N995 #800 = 5;
N995 #1105 = 1;
GOTO 50;
```

Dans la partie IF, le (#600 EQ 45) permet de vérifier que dans l'étape précédente (l'étape de cadrage), le vérin correspondant est bien passé à 45°, le vérin de cadrage est lié à la variable #600 et à un tenkey sur le fanuc picture. Le (#700 EQ 1) est une variable liée au code M permettant de bouger le vérin et elle est de base à 1, grâce à cette variable, les programme ne reste pas bloqué dans la boucle.

```
; (#Av5 - SERTISSAGE)
N50 IF[#1006 EQ 1] AND [#605 EQ 45] AND [#800 EQ 5]] GOTO 994;

(SERTISSAGE)
N994 #1105 = 0;
N994 #800 = 6;
N994 #1106 = 1;
GOTO 60;
```

Dans la partie IF, le (#605 EQ 45) permet de vérifier que dans l'étape précédente (l'étape de serrage), le vérin correspondant est bien passé à 45° (pour cette partie on choisit 45° de manière arbitraire).

Dans le code, on reset le compte rendu, on augmente la valeur de #800 pour correspondre à l'étape et on passe le F correspondant à 1.

```
; (#Av6 - Déchargement par convoyeur)
N60 IF[#1007 EQ 1] AND [#610 EQ 4] AND [#800 EQ 6]] GOTO 993;

(CONVOYEUR)
N993 #1106 = 0;
N993 #1107 = 1;
GOTO 70;
```

Dans la partie IF, le (#610 EQ 4) permet de vérifier que dans l'étape précédente (l'étape de sertissage), les 4 mâchoires ont bien effectué le sertissage.

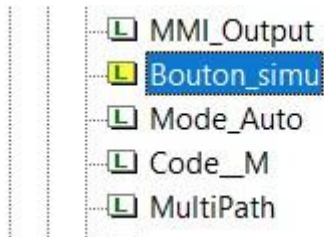
Dans la fonction, le #3006=30 permet d'envoyer un message qui n'arrête pas le programme.

```
N70 IF[#1008 EQ 1] GOTO 992;  
;  
(RESET)  
N992 G00 X-75 Y-305 Z0; (RESET POS)  
N992 #1103 = 0;  
N992 #1107 = 0;  
N992 #700 = 1;  
N992 #600 = 0;  
N992 #605 = 0;  
N992 #610 = 0;  
N992 #800 = 0;  
GOTO 80;
```

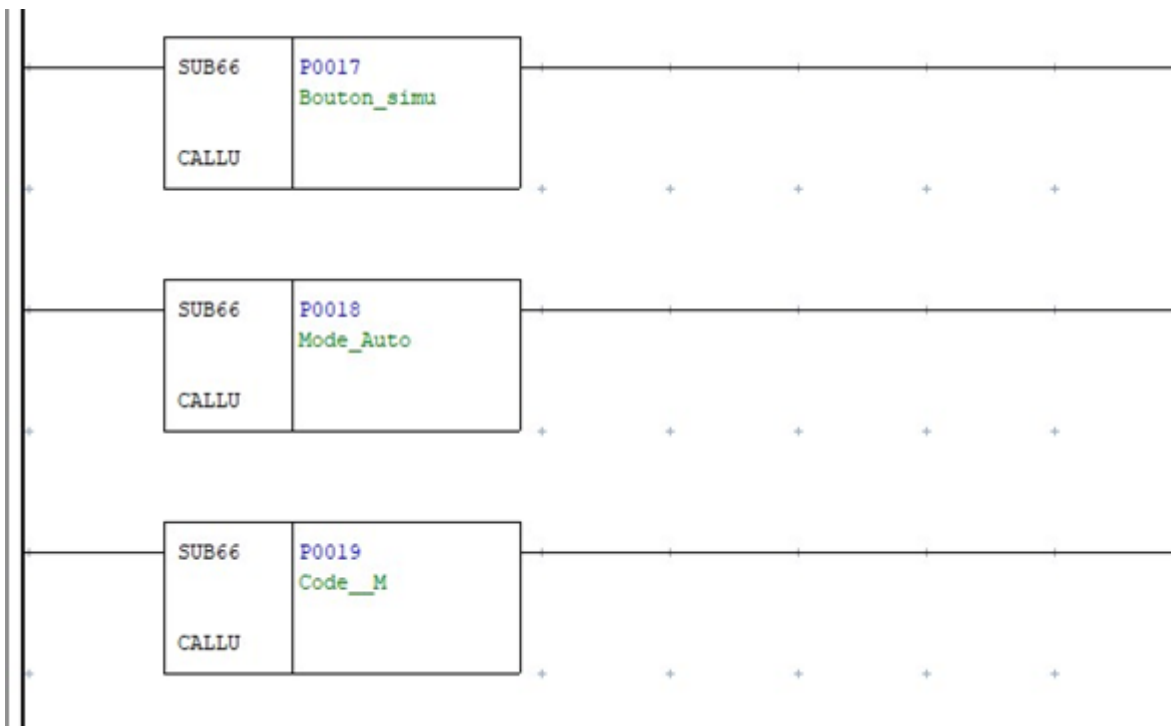
Dans cette partie du programme, on reset toutes les variables afin de pouvoir relancer un nouveau cycle en repartant de 0.

5) Mission 5 :FANUC LADDER

Dans un premier temps, nous avons donc récupéré la PMC déjà présente sur la machine afin de la modifier pour y ajouter les capteurs que nous devons simuler, le nouveau mode automatique mais aussi le code M pour l'étape du serrage de la fenêtre par les vérins verticaux. Nous avons donc ajouté en tout trois sous-programmes dans la PMC de base de la machine qui sont les sous-programmes numéros 17, 18 et 19 :

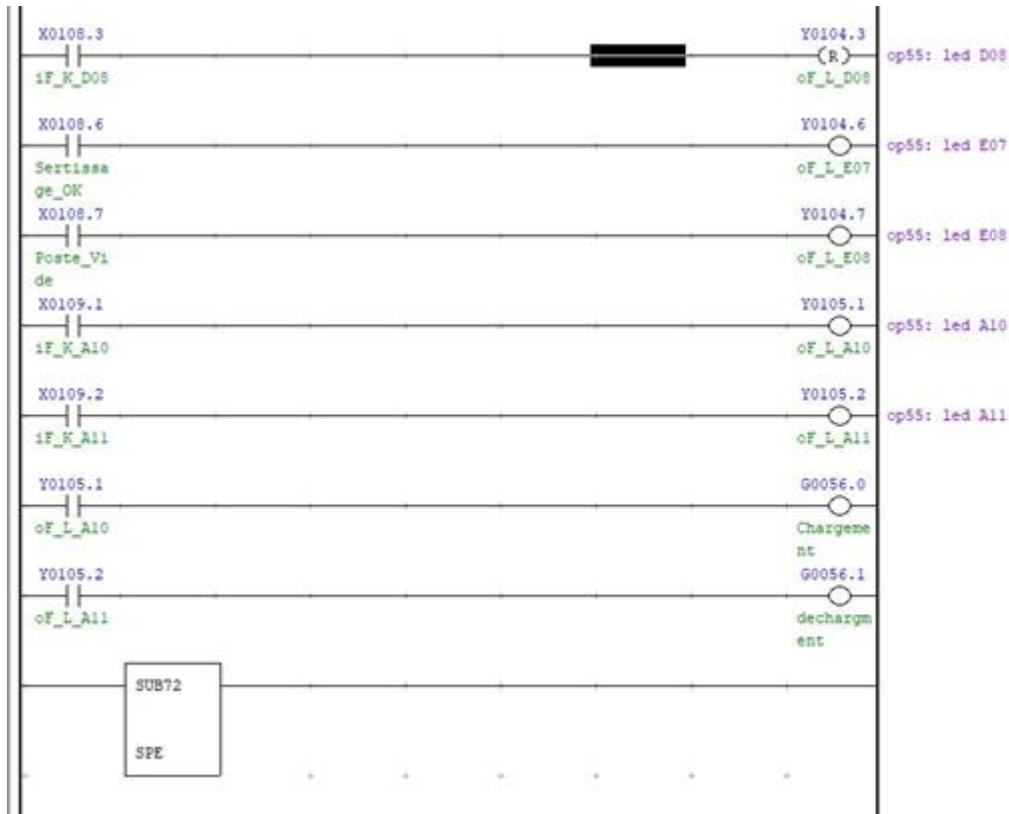


Nous avons déclaré ces nouveaux sous-programmes dans la partie LEVEL 2 pour bien faire appel à ces sous-programmes :

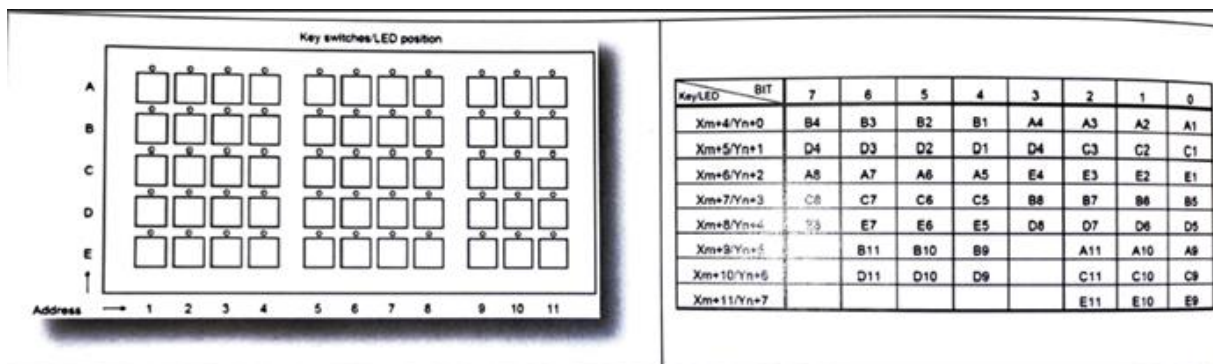


Page simulations des capteurs :





Nous avons donc simulé l'utilisation des capteurs par des boutons présents sur le clavier qui activent les LEDs qui leur correspondent comme vous pouvez le voir un X (bouton) active un Y (LED). Afin de s'assurer que nous prenons bien les bons boutons et les bonnes LEDs que nous voulons utiliser nous utilisons le tableau suivant qui nous donne toutes les positions des boutons et des LEDs avec les adresses qui leur correspondaient :



Nous avons donc déclaré chaque nouveaux capteurs (boutons) dans la table des matières présente dans FANUC LADDER afin que notre programme soit lisible par chacun :

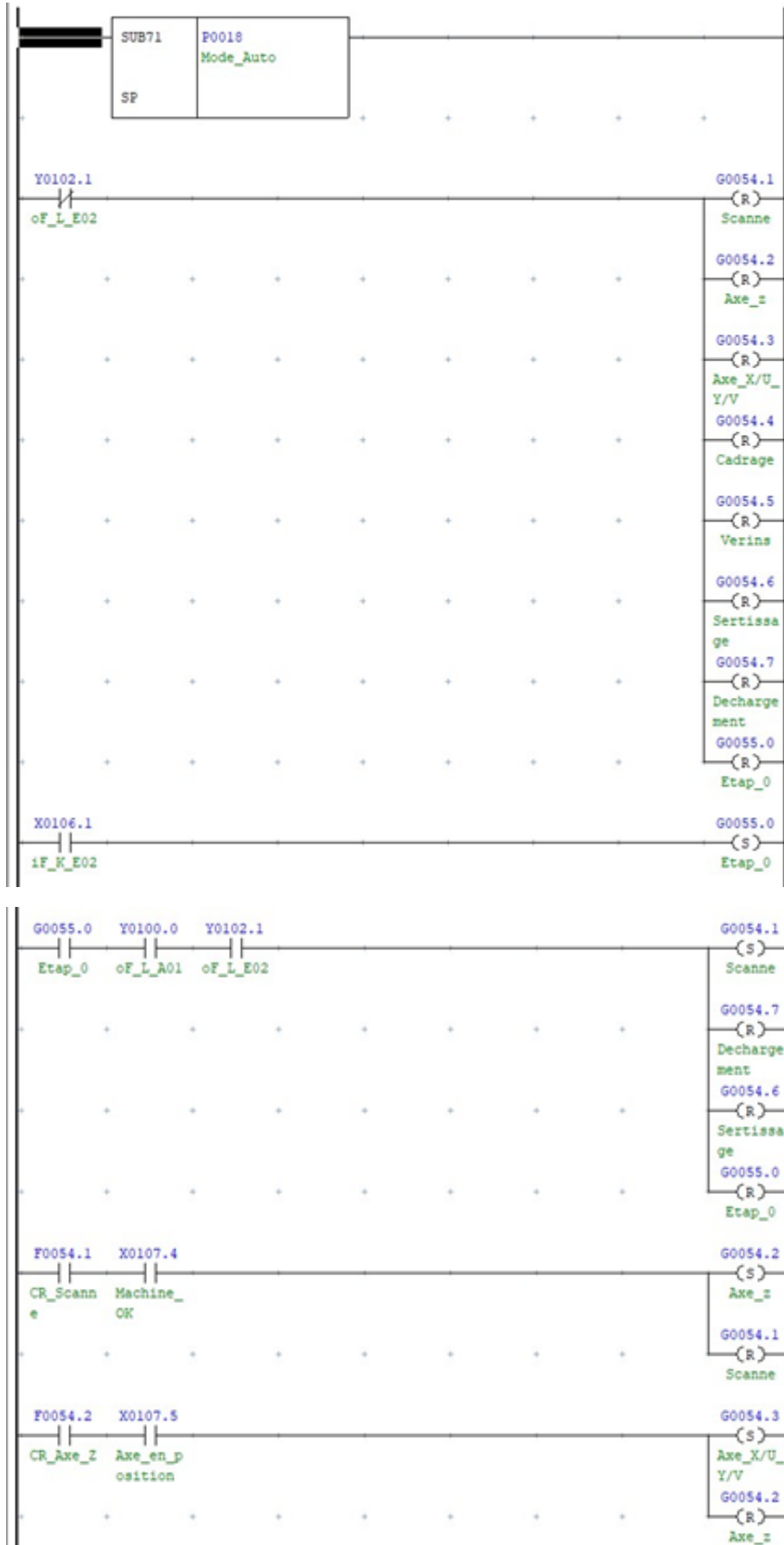
860	X0107.3	iF_K_B08	op55: key B08
861	X0107.4	Machine_OK	op55: key C05
862	X0107.5	Axe_en_position	op55: key C06
863	X0107.6	Position_OK	op55: key C07
864	X0107.7	Cadrage_OK	op55: key C08
865	X0108.0	iF_K_D05	op55: key D05
866	X0108.1	iF_K_D06	op55: key D06
867	X0108.2	iF_K_D07	op55: key D07
868	X0108.3	iF_K_D08	op55: key D08
869	X0108.4	Position_Fenetre	op55: key E05
870	X0108.5	Serrage_OK	op55: key E06
871	X0108.6	Sertissage_OK	op55: key E07
872	X0108.7	Poste_Vide	op55: key E08
873	X0109.0	iF_K_A09	op55: key A09
874	X0109.1	iF_K_A10	op55: key A10
875	X0109.2	iF_K_A11	op55: key A11

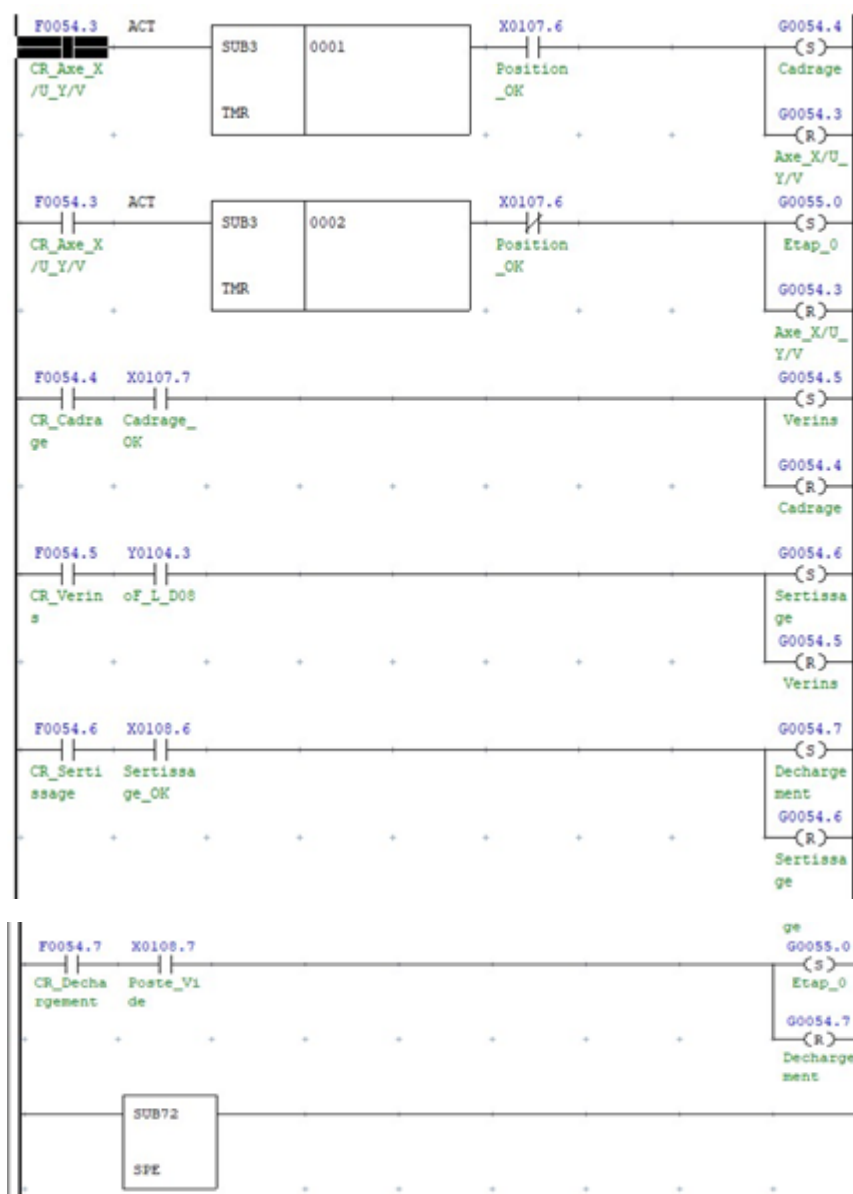
Dans le cas où deux capteurs doivent être actifs pour passer à l'étape suivante du mode automatique (passage à l'étape du sertissage par 4 mâchoires horizontales), lorsque le X d'un des deux capteurs est actif, celui-ci fait un set sur le Y de la LED qui lui correspond. Si les deux LEDs des deux capteurs (Serrage_Ok et Position_Fenêtre) sont actifs alors on reset les deux LEDs des deux capteurs et une autre LED, la LED Y104.3 (D08) du clavier fait un set jusqu'à temps que l'on appui sur le X108.3 pour désactiver celle-ci.



Nous avons pour finir deux derniers capteurs qui doivent montrer que la fenêtre est soit au poste de chargement ou de déchargement afin de remonter cette information à la CNC et à l'IHM, l'activation de ces deux capteurs se fait par un bouton (X) active une LED (Y) mais aussi un code G qui servira à remonter l'information à la CNC et à l'IHM.

Page mode automatique :





Tout au début du programme automatique, on fait un reset de tous les codes G des différentes étapes si la LED du cycle start n'est pas active. Ce reset nous permet en cas de dysfonctionnement du mode automatique de reset toutes les actions du mode.

Pour activer le mode automatique, il faut que l'étape 0 qui est le G55.0 soit actif et pour qu'elle soit active, il faut que le bouton cycle start soit actif. Pour lancer la première étape du mode automatique, c'est-à-dire celle du scanne du produit, il faut que la LED du mode automatique et du cycle start soient actives.

Les différentes étapes du mode automatique sont déclarées comme des codes G et les comptes-rendus de ces étapes sont déclarés en tant que code F. Nous avons déclaré ces différentes étapes et leurs comptes-rendus dans la table des matières du FANUC LADDER que vous pouvez voir ici:

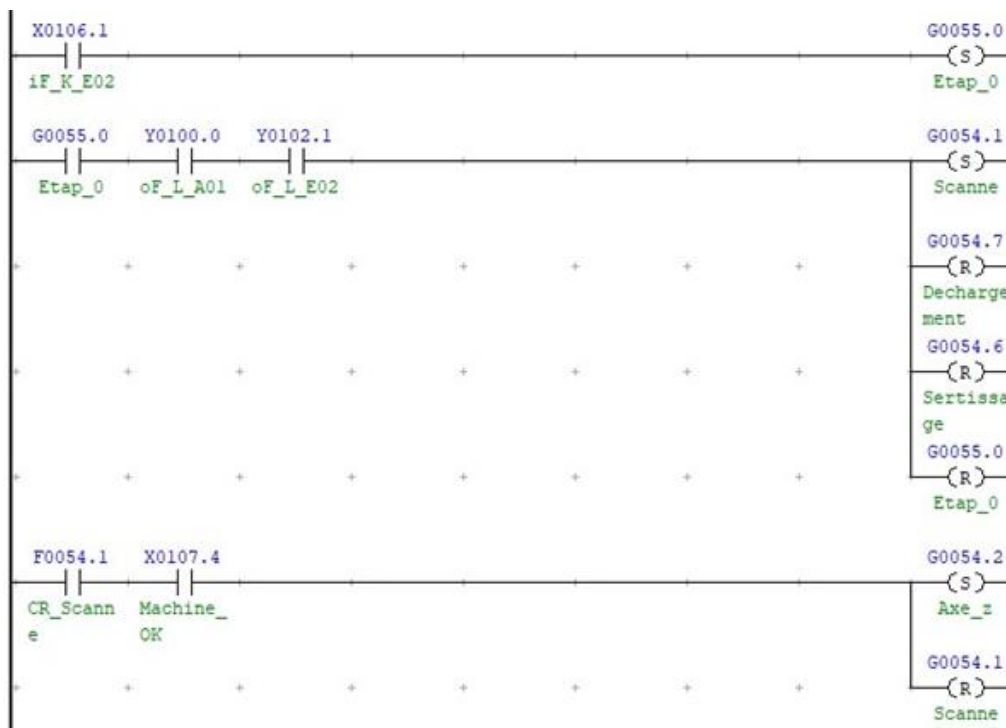
Table des matières des codes F :

434	F0054.1	CR_Scanne
435	F0054.2	CR_Axe_Z
436	F0054.3	CR_Axe_X/U_Y/V
437	F0054.4	CR_Cadrage
438	F0054.5	CR_Verins
439	F0054.6	CR_Sertissage
440	F0054.7	CR_Dechargement

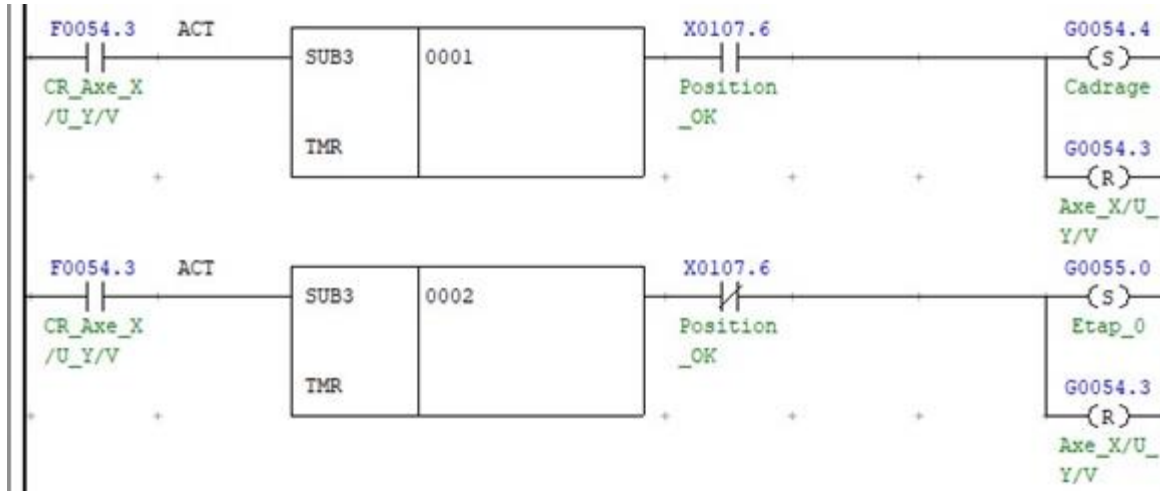
Table des matières des codes G :

434	G0054.1	Scanne
435	G0054.2	Axe_z
436	G0054.3	Axe_X/U_Y/V
437	G0054.4	Cadrage
438	G0054.5	Verins
439	G0054.6	Sertissage
440	G0054.7	Dechargement
441	G0055.0	Etap_0

Ces codes G et F vont nous permettre de communiquer de la PMC à la CNC. En effet lorsqu'un code G va être actif dans la PMC une variable de la CNC va être mis à 1 et celui-ci va donc pouvoir être tester avec un IF dans le code ISO afin de réaliser la tâche qui est à effectuer après cette tâche le code ISO va nous renvoyer un compte-rendu F en mettant une variable à 1 qui va pouvoir être alors testé par le ladder de la PMC . Donc, pour passer à une étape suivante du mode automatique, nous allons tester si le compte-rendu de l'étape précédente est actif et que le capteur ou les capteurs qui correspond/ent au passage à l'étape suivante est/sont aussi actif/s et ainsi de suite pour chaque étape du mode automatique.



Par exemple dans ce cas, le G54.1 va mettre à 1 le #1001 qui va être testé dans le code ISO qui après avoir effectué la tâche du scan va mettre à 1 le #1101 qui pourra être testé dans la PMC comme le compte-rendu du code G54.1 qui sera le F54.1 une fois que le compte rendu est reçu on attend que le capteur pour passer à l'étape suivante soit actif. Une fois qu'il est actif on set l'étape d'après qui est l'axe Z (G54.2) puis on reset l'étape d'avant qui est le scanne (G54.1) est ainsi de suite.



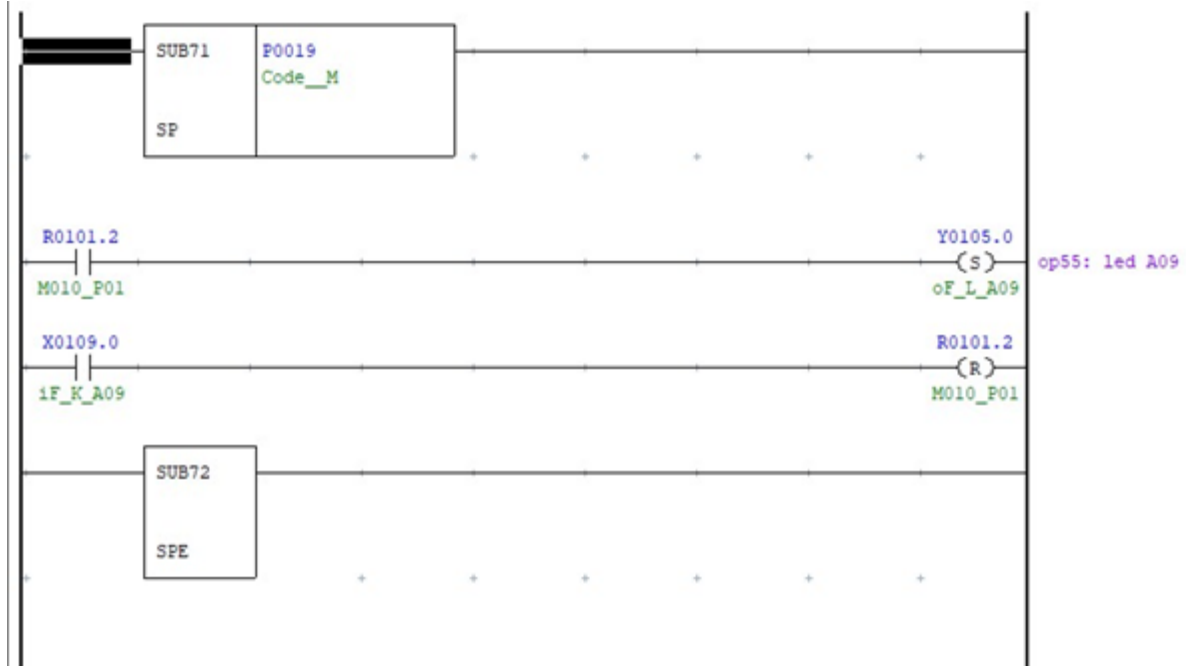
Dans le cas où nous devons tester si la position est OK ou non, nous avons utilisé les timers afin de laisser le temps au code ISO d'effectuer la tâche du déplacement des axes X/U et Y/V et que le capteur ait le temps d'être actif ou non. S'il est actif, on passe à l'étape suivante G54.4 s'il n'est pas actif donc la position n'est pas ok, on retourne on set l'étape 0 et donc par conséquent on reset le positionnement des axes de la machine et on retourne à l'étape du scan.

Pour utiliser les timers on a utilisé les sub3, on a pris les timers 1 et 2 que l'on a configuré de manière à générer une temporisation de 10 secondes (diagnose-> PMC parameter-> timer)



Après que la dernière étape du déchargement soit effectuée et que l'on ait reçu son compte-rendu et que le capteur poste vide soit actif, on retourne à l'étape 0 (reboilage) et on effectue donc un reset des axes de la machine avant de retourner à l'étape du scan.

Page code M :



Ce sous-programme sert à réaliser le code M de l'étape du serrage de la fenêtre par les vérins verticaux.

Lorsque le code M10 est actif dans le code ISO, le R101.2 qui correspond à ce code M est actif au niveau de la PMC. On test alors ce R dans la PMC et si celui-ci est actif alors on set la LED A09 sur le clavier de la machine ici la LED Y105.0 et lorsque que le M10 a fini de réaliser sa tâche on appui sur le X109.0 qui correspond au bouton correspondant à la LED pour reset le R101.2 afin de pouvoir continuer la suite du code ISO et dans le même temps on reset la LED .