E700 RESUME INSTRUCTIONS UNIPROG

Notations : dst : Un argument noté dst signifie que l'opération va modifier sa valeur.

En gris clair, les opérations sont réservées au système ou pas encore implémentées.

En gris foncé et en italique, les instructions provenant de l'ancien UNIPROG E600

Entre crochets [ ] argument optionnel présent zéro ou une seule fois.

Entre accolades { } argument optionnel présent zéro, une ou plusieurs fois.

ABS dst	Valeur absolue : dst ← |dest|
ADD dst src	Addition : dst ← dst + src
ADDD src	*Addition directe : accum ← accum + src*
ADDT Fadr src	Add Time. Fadr est une adresse FRAM. scr est un nombre de secondes. #FRAM[Fadr] est le résultat en heures, #FRAM[Fadr+1] en minutes et #FRAM[Fadr+2] en secondes.
AND dst src	ET «bitwise» : dst ← dst AND src (& bit à bit : 1 & 2 = 0)
ASIM [src] label	Activation d'une tâche simultanée
ATN dst	Arc tangente : dst ← arc tg dst
BRIN0 src label	Branche à label si src = 0
BRIN1 src label [src label]	Branche à label si src ≠ 0 [active une tâche sim. (ASIM)]
BRKPT	Breakpoint (réservée au système)
BRM label	*Branch if minus : Saute à label si accum < 0*
BRNZ label	*Branch if non zero : Saute à label si accum ≠ 0*
BRP label	*Branch if plus : Saute à label si accum > 0*
BRZ label	*Branch if zero : Saute à label si accum = 0*
CALIN0 src label { parami }	Appelle la procédure label si src = 0. Paramètres optionnels
CALIN1 src label { parami }	Appelle la procédure label si src ≠ 0. Paramètres optionnels
CALL label { parami }	Appelle la procédure label. Paramètres optionnels
CALLSI label	Appelle la procédure label. Réservé au système pour les sous-programmes ISO.
CASE src label1 label2	SWITCH CASE ENDS : Si valeur = src, alors appelle la procédure label1. Retour à label2
CINR dst	Conversion Integer Round : dst ← Arrondi (dst)
CINT dst	Conversion Integer Trunc : dst ← Partie entière (dst)
*CIRA axe1 coord1axe2 coord2 [cmx cmy mode]	Arc de cercle, coordonnées absolues
*CIRR axe1 coord1axe2 coord2 [cmx cmy mode]	Arc de cercle, coordonnées relatives
CLR232	Nettoyage du tampon de réception RS232 (si switch 5 ON)
CLRORG sim [espace]	Mise à 0 des origines (G53 + T-1 + G60 D-1 + G59). Si espace omis, c'est l'espace ISODEF qui est concerné.
CMP src1 src2	Comparaison de src1 et src2 (src1 – src2)
COS dst	Cosinus : dst ← cos dst
CPL dst	Complément à 1 : Si dst = 0 alors dst ← 1; sinon, dst ← 0
CYCLN sim msg	Exécute le programme msg dans la tâche sim. msg est un programme ISO et si sim = -1 c'est le power on.
DEC dst	Décrémente : dst ← dst - 1
DISPC src { srci }	Affiche le caractère dont le code ASCII est src. 0=Clear Sreen, 1=Clear Line, 2=Del begin of line, 3=Del end of line, 7=Beep, 8=Backspace, 127=Del character.
DISPN src [digits] format	Affiche la valeur de src avec format chiffres après la virgule. digits = taille totale (part. entière + part. frac.)
DISPS src	Display Screen Affiche l'écran numéro src
DISPST msg	Display String : Affiche la chaîne numéro msg (Attention : Pas de # dans les msg)

DIV dst src	Division : dst ← dst / src
DIVD src	*Division directe : accum ← accum / src*
DPATH groupe espace vitesse	Définition d'une interpolation
END	Fin de tâche ou de procédure
ENDP	Fin d'une interpolation DPATH… ENDP
ENDRP	Fin d'une boucle REP … ENDRP
ENDS	Fin d'un SWITCH/CASE : SWITCH CASE { CASE } ENDS
ERROR msg mode	Erreur avec affichage du message. Si mode = 0, arrêt total. Sinon c'est une mise en pause avec possibilité de continuer. Ne pas utiliser en mode 1 dans l'AUTOMAT !
G5358 src [sim]	Exécution d'un Gsrc. Préciser sim si espaces multiples. Sinon sim = 0.
*GETKF dst	Attente sur une touche F1..F6.
GTXY srcx srcy	Positionne le curseur
ICNT	Pas implémentée !
INC dst	Incrémente : dst ← dst + 1
INP dst up down precis	Saisie d'un nombre down ≤ src ≤ up avec precis chiffres après la virgule. **Bloque tout** !
*INP1 [dst posX posY fl fr min max mode]	Saisie d'un nombre dans dst. posX et posY sont les positions du curseur. fl est le nombre de chiffres à gauche de la virgule, fr, le nombre de chiffres à droite. min et max sont les bornes et mode = 0 normal ou mode = 1 pour effet «stabilo». Si pas de paramètres, permet de quitter soft.
INV dst	Inversion : dst ← 1 / dst
(**)ISO msg	Exécution style IMD d'une ligne ISO. La ligne ISO est dans MSG.INI. Ne s'exécute pas depuis un sim > 4.
ISODEF sim msg espace [axeX [axeY [axeZ]]]	Définition d'un ISO multiple. Sim est le numéro de la tâche, msg est le nom du fichier ISO et espace est le groupe d'axes. Optionel : Redéfinition des axes possible pour G17 et utiliser XYZ au lieu des autres noms (seul. en ISO)
ISORUN sim	Exécution de l'ISO précédemment défini.
JE label	Jump if Equal : après un CMP src1 src2, saut si src1 = src2
JG label	Jump if Greater : après un CMP src1 src2, saut si src1 > src2
JGE label	Greater or Equal : après un CMP src1 src2, saut si src1 ≥ src2
JL label	Jump if Less : après un CMP src1 src2, saut si src1 < src2
JLE label	Less or Equal : après un CMP src1 src2, saut si src1 ≤ src2
JMP label	Jump : Saut inconditionnel à label
JNE label	Jump if not Equal : après un CMP src1 src2, saut si src1≠src2
KSIM src	Kill simultaneous : Tue la tâche numéro src
*LINA axe coord mode	Segment absolu. mode 0 : préparatoire. Sinon exécution. Utilisée pour mélanger circulaire et linéaire (mode 0)
*LINA2 ax coord { axe coord }	Segment absolu, écriture simplifiée pour plusieurs axes.
*LINR axe coord mode	Segment relatif. mode 0 : préparatoire. Sinon exécution. Utilisée pour mélanger circulaire et linéaire (mode 0)
*LINR2 ax coord {axe coord }	Segment relatif, écriture simplifiée pour plusieurs axes.
LOAD src	*Chargement : accum ← dst*
LOG [src]	Mémorise src et le numéro de tâche dans le fichier log. Si src est ommise, c'est le numéro de ligne qui est mémorisé. **Attention**, src est enregistrée en entier (pas de décimales) !
MAP sim axeX [axeY [axeZ]]	Redéfinition des axes possible pour G17 et utiliser XYZ au lieu des autres noms (seul. en ISO). C'est comme ISODEF mais simplifié. Probablement inutile !
MEMR type dst [1]	Memory Read. Lit toute la mémoire du E700.Auto-incrémental type = 1, 2, 3 ou 4 pour 1, 2, 3 ou 4 bytes type = 5 pour les réels type = 6 pour lire l'adresse de la base de la RAM externe libre (remote seulement) type = 7 pour lire l'adresse de du top de la RAM externe libre (remote seulement) Si l'argument 1 est présent, on lit dans le remote **Pas de compatibilité entre les différentes versions !**

MEMW type src [1]	Memory Write. Ecrit toute la mémoire du E700. Auto-Incrémental. **Cette instruction est dangereuse. Pas de compatibilité assurée entre les différentes versions !**
MENU f m [s]	Affichage d'un menu f fonction (F1 à F6) –1 pour supprimer la fonction. m message en provenance de msg.ini. [s] Numéro d'écran dans lequel on change. Si ce paramètre est omis, on change dans l'écran courant.
MFILE	
MOD dst src	Reste de la division entière : dst ← dst MOD src
MOV dst src	Affectation : dst ← src
*MSG msg	Affichage du message msg avec demande de quittance(EXIT) Quittance soft si msg = -1. La quittance soft ne fonctionne que si la tâche en attente sur MSG n (n ≥ 0) a préalablement été tuée ! Possibilité d'utiliser des #.
MUL dst src	Multiplication : dst ← dst * src
MULD src	*Multiplication directe : accum ← accum * src*
NEG dst	Négation : dst ← (-1) * dst
NOP	No operation. Utile pour remplir un label
NOT dst	Complément à 1, bit à bit d'un entier : dst ← ~dst (~1 = -2)
OFF dst	Reset : dst ← 0
ON dst	Set : dst ← 1
OR dst src	OU «bitwise» : dst ← dst OR src (| bit à bit : 1 | 2 = 3)
**PECK mode axe profondeur tempo profondeurDePasse garde vitesse	Cycle de perçage. Voir annexe.
POP dst	Dépile
*POSA axe vitesse pos mode	Positionnement absolu. Mode 0 préparatoire, 1 pour exécution sur un seul axe, 2 pour exécution sur tous les axes préparées et 3 pour positionnement sans attente.
*POSO axe vitesse pos mode	Positionnement absolu par rapport au origines machine (REF)
*POSR axe vitesse pos mode	Positionnement relatif.
PSIM [sim]	Met en pause toutes les tâches, sauf celle qui appelle. Si sim est précisé, seule la tâche sim est mise en pause avec en plus, les mouvements sont mis en pause. On l'utilise avec un paramètre principalement pour implémenter un bouton pause en ISO multiple.
PUSH src	Empile
*QREF axe	Référence rapide sur index seulement.
RAD src mode	Rayon d'arc interpolé. Mode 1=clockwise,0=counterclockwise
**REF axe	Prise de référence. Utilise le timer seulement en mode absolu (si rcoder non nul dans E700.INI)
*REFR axe retour	Prise de référence sur un axe avec règle avec retour rapide
REP src	Boucle répétée src fois (REP src …. ENDRP)
RSIM [sim]	Réactivation des tâches mises en pause par PSIM
RX	RS-485. Pas encore implémenté
**RX232 l/c adresse	Réception RS232 (switch 5 ON). l/c : longueur ou code du caractère terminal (choix dans RXOPT); adresse du tableau (donc pas de # !)
SAVE	Exécution d'une sauvegarde. **Utiliser avec parcimonie pour éviter un vieillissement prématuré de la flash sérielle**.
*SCUT src1 { srci }	Shortcut. Touches raccourcis. Voir feuilles annexes.
SHL dst src	Shift left : dst ← dst SHL src
SHR dst src	Shift right : dst ← dst SHR src
SIN dst	Sinus : dst ← sin dst
*SKS src1 { srci }	Short Key sequence. Pour «presser» sur des touches en soft

SPEC	src …	Fonctions spéciales
	0 Encodeur	***0 axe encoder p2*** : Affecte la valeur de encoder modulo le nombre de pas par tour de l'axe à pabs [axe]. Attention, le nombre de pas par tour doit être une puissance de 2 (p2) !!!
	1 Stabilo	***1 ligne début fin mode*** : Si mode = 1 : Passe la ligne au stabilo depuis début (de 0 à 41) jusqu'à la fin. Si mode=0: Annulation.
	2 Affiche programme	***2 x y mode*** : Si mode = 0, affichage en *x*; *y* du nom du power on program; mode = 1 : cycle program et mode = 2 : 1ère ligne du cycle program (commentaire)
	****3** Test **Ethernet**	***3 code*** : Teste présence Ethernet. Ok si code retourné = 0.
	****4** Write **Ethernet**	***4 no type val code*** : Ecrit la valeur val dans la variable Eternet numéro no. Type 0 = entier, 1 = réel (3 décimales) et 2 = booléen. Code retourné = 0 si communication ok !
	****5** Sauve **Ethernet**	***5 type acces code*** : Sauvegarde des valeurs Ethernet dans un fichier (dans le module Ethernet). Type 0 = format XML ou 1 = format CSV. Acces = 0 pour Write (Rewrite) et bousiller la dernière sauvegarde ou acces = 1 pour concaténer (append). Attention, place limitée ! Code retourné = 0 si ok.
	6 Lire ligne 1	**6 adr long** : Lit la première ligne du programme cycle et stocke les long premiers caractères à l'adresse adr (donc pas de # !)
	7 FRAM→FUSER.INI	**7 nbre** : Permet de créer un fichier FUSER.INI contenant les nbre premières variables de la FRAM.
	8 FUSER.INI→FRAM	**8** : Permet de copier les valeurs de FUSER.INI dans la FRAM.
	9 Modif. TOOL POS	**9 n° axe** : (n° : de 0 à 3) Modifier la liste dans TOOL POS.
	10 Temps d'exécution R	**10 p_1 p_2 ... p_n** : Rien ! Juste pour calculer les temps d'exécution avec arguments en lecture.
	11 Temps d'exécution W	**11 p_1 p_2 … p_n** : Rien ! Juste pour calculer les temps d'exécution avec arguments en écriture.
	12 AE2 UART test	**12 axe** : Pour tester communication UART avec Yaskawa. Résultat du test dans UARTERR.
	****13** Remlog Ethernet	**13 typ val code** Envoi de message sur Ethernet. typ 0=entier, 1=réel (3 décimales), 2=boolean et 3=string(msg.ini). Code retourné = 0 si communication ok !
	****14** Remlog Ethernet	**14 access code**. 0=open, 1=close. Code : voir 13 ci-dessus.
	15 axe val	**15** axe val : FPABS[axe] ← FPABS[axe] + val. Les STROKEs sont aussi modifiés. SPEC 15 annule le dernier SPEC 15.
	16 axe1 axe2	Echange de deux noms d'axes
	17	Ecriture de UNIVS[0..49] dans nnn.TXT (nnn = #FILENAME)
	18	Lecture (inverse de 17)
	19 bout coul	Colorie en coul le bouton bout (TouchScreen uniquement)
	20	
SPINDL	mode	POT spindle, broche, entrées/sorties analogique 0 pour l'affecter au DAC 0 1 pour l'affecter au DAC 1 2 pour l'affecter aux DAC 0 et 1 4 pour le désactiver 5 axe pour l'affecter à spli [axe] 6 axe pour le désactiver de spli [axe] 7 pour l'affecter à SPOT [0..255]
SQR	dst	Elévation au carré : dst ← dst^2
SQRT	dst	Racine carrée : dst ← $dst^{1/2}$
START	dst	Etat du bouton START : dst ← état selon la configuration
STOPM	axe mode	Stop motion. Arrêt de l'axe (tous les axes si axe < 0) mode = 0 avec rampes et mode = 1 sans rampes
STORE	*dst*	*Inverse de LOAD : dst ← accum*
SUB	dst src	Soustraction : dst ← dst - src
SUBD	*src*	*Soustraction directe : accum ← accum - src*
SWITCH	src	SWITCH CASE ENDS

TAN	dst		Tangente : dst ← tg dst
*THREAD		pas pro prf ang fin n	Filetage : pas, profondeur de filetage, profondeur de la passe de finition, angle, position Z de la fin de filetage, nombre de passes.
TOOL	src [sim]		Outil G60 Dsrc. Préciser sim si espaces multiples. Sinon sim = 0
TOOLI	src [sim]		TOOL ISO : Tsrc. Préciser sim si espaces multiples. Sinon sim = 0.
*TPING	axe pas coord sortie1 [S2]		Taraudage sur porte-outil souple axe : Axe vertical pas : En mm coord : Comme dans POSA sortie : Sortie à inverser pour la remontée S2 : Seconde sortie à inverser en cas de besoin.
*TX	module index valeur err		Envoie au module, à l'index la valeur. Retourne err = 0 si ok et un code d'erreur sinon.
*TX232	longueur adr/msg		Transmission RS232 (si switch 5 ON). Adresse du tableau (donc pas de # !) ou numéro du message dans MSG.INI. Choix dans TXOPT.
(**)UART		src …	Gestion de la communication sur nouvelles cartes AE
		0 ser baudrate parité data stop	Initialisation de la communication. Parité N=0, 1=Odd et 2=E.
	(*)	**1** ser longueur adr/msg	Envoi d'une chaîne de caractères. Voir UTXOPT et TX232. Atomique, mais utilise le timer !
	**	**2** ser l/c [adresse]	Réception d'une chaîne de caractères. Dans USTRREC si adresse omise. Voir URXOPT et RX232.
		3 ser	Effacement du tampon de réception ser est le numéro du port sériel (pas de l'axe !)
**WAIT	src		Attente de src secondes
WAIT0	src		Attente si src = 0
WAIT1	src		Attente si src ≠ 0
WAITK	dst		Attente sur une touche. Code dans dst. **Bloque tout** !
XOR	dst src		Ou exclusif «bitwise» : dst ← dst XOR src (^ bit à bit)

Note importante : Il n'existe pas de correction d'outil en UNIPROG (G40, G41, G42). Il y a toutefois la possibilité d'appeler un programme ISO depuis UNIPROG. Ceci permet donc de contourner ce fait.

***

* Instruction non-atomique
** Instruction non-atomique utilisant le timer