



AUTOMATISME POUR LA ROBOTIQUE

TP 3

v0.2

IUT d'Annecy, 9 rue de l'Arc en Ciel, 74940 Annecy

GMMA ET PREMIERS MOUVEMENTS

1 Programmation du robot via uniVALplc



Principe de la programmation robot

Dans notre structure, le robot est piloté par l'automate M340 via l'API uniVALplc. L'envoi des ordres de mouvements se fait d'une file de commande de mouvement (FIFO - First In First Out). Les commandes seront effectuées par le robot dans l'ordre reçu. Chacune d'entre elle sera retirée de la file une fois exécutée.

Pour rappel :

- L'exécution séquentielle des commandes de mouvement dans la file débute dès que le bras est mis sous puissance (MC_GroupPower).
- Il est possible d'ajouter des commandes de mouvement à tout moment, y compris lorsque le robot est en mouvement ou hors puissance.
- Une commande de mouvement sort de la file dès que sa trajectoire associée est totalement achevée.
- Pour vider la FIFO de toutes les commandes de mouvement empilées, on dispose du bloc fonctionnel MC_GroupStop.
- L'arrêt immédiat du robot durant l'exécution d'une trajectoire peut se faire
 - ◊ via le bouton d'arrêt d'urgence (E-Stop)
 - ◊ par un changement de mode de marche sur la baie (WMS9)
 - ◊ à l'aide des blocs fonctionnels MC_GroupInterrupt et MC_GroupStop.
- Suite à un arrêt obtenu à l'aide du bloc fonctionnel MC_GroupInterrupt, il est possible :
 - ◊ de reprendre les mouvements là où ils ont été stoppés grâce au bloc fonctionnel MC_GroupContinue
 - ◊ d'interrompre les mouvements en vidant la file (MC_GroupStop vu précédemment).
- Une mise sous puissance du bras conduira éventuellement à une phase dite de « connexion » faite à petite vitesse pour rejoindre la trajectoire interrompue. Les mouvements toujours présents dans la file seront ensuite repris. Il est donc important de vider la file avant de mettre le bras sous puissance si l'on souhaite éviter ce comportement.

Le tableau ci-dessous présente les blocs fonctionnels de l'API uniVALplc permettant de mettre sous puissance et la gestion des erreurs du robot.

Name	Description
MC_GroupInterrupt	Stop the movement of the robot. The robot can continue along the commanded path after MC_GroupContinue is executed
MC_GroupStop	Stop the movement of the robot and flushes all movements that have been commanded
MC_GroupContinue	Restart the movement of the robot after it has been stopped by MC_GroupInterrupt
MC_GroupReset	Reset all pending errors reported by the robot
MC_GroupPower	Switch ON/OFF the power stage of the robot



Téléversement d'un programme

Lorsque l'on télécharge un nouveau programme dans l'automate, la communication entre l'API M340 et la baie CS9 ainsi que celle avec le M221 sont rompues. Il en est de même avec le PFC200. Les bits de vie entretenus par ces communications ne s'effectuant plus :

- la baie CS9 passe en défaut, interdisant toute commande sauf naturellement celle qui permet d'acquitter ces erreurs (MC_GroupReset).
- Le M221 est programmé pour faire clignoter à la fréquence de 1 Hz la composante rouge de la colonne qu'il gère.
- Le PFC200 fait passer le poste Guichet en **non passant** et sa colonne prend la couleur magenta.

Pour la remise en service :

- Les deux contrôleurs M221 et PFC200 retrouvent leur fonctionnement normal dès que la communication devient opérationnelle.
- Il faut mettre en place une procédure de redémarrage pour la baie CS9 (cf. 2.2).

Les commandes de mouvement de type **PLC-Open** disponibles sont les suivantes :

Nom du bloc fonctionnel	Description	Type de mouvement
MC_MoveAxisAbsolute	Joint interpolated movement	joint position (T_JointPos)
MC_MoveDirectAbsolute	Joint interpolated movement	cartesian position (T_CartesianPos)
MC_MoveLinearAbsolute	Linear movement	cartesian position (T_CartesianPos)
MC_MoveCircularAbsolute	Circular movement	cartesian position

Préparation 1 : Types de mouvement

A partir de la documentation du client **uniVALplc** fournie dans le sous-dossier **Scara/uniVALplc/Doc/** :

Question 1 Expliquer le fonctionnement de chacun des blocs fonctionnels du tableau ci-dessus.

1.1 Bonnes pratiques de programmation

L'utilisation des blocs fonctions en ST pour la commande du robot implique une certaine rigueur dans la programmation. En effet, par le nombre de paramètres qu'ils nécessitent, l'appel d'un bloc est particulièrement volumineux en terme de nombre de lignes de codes. Cela complexifie grandement la lecture du programme s'ils sont insérés directement au moment de leur utilisations.

Ainsi, nous proposons de déplacer l'appel des blocs fonctions dans une section (sous-programme) dédiée. On leur associera une variable structurée qui contiendra l'ensemble des paramètres nécessaires à leur usage. Leur déclenchement se fera alors au travers de la variable booléenne d'activation, présente dans la structure.

L'encart ci-dessous propose une démarche pour la mise en oeuvre de cette méthode.



Utilisation des types structurés

Pour chaque type de bloc fonctionnel, on créera un type structuré associé. Ce type structuré contiendra l'ensemble des paramètres nécessaires à l'appel du bloc fonctionnel. Pour éviter la multiplication du nombre de types différents, on propose la mise en commun suivante :

Type de fonction	Type associé	Blocs concernés
Mouvements du robot	T_MoveAxisAbsolute	MC_MoveAxisAbsolute MC_MoveDirectAbsolute MC_MoveLinearAbsolute MC_MoveCircularAbsolute
Gestion globale du robot	T_ctrlStatusGroup	MC_GroupInterrupt MC_GroupStop MC_GroupContinue MC_GroupReset MC_GroupPower

Les types associés contiendront l'ensemble des paramètres nécessaires à l'appel de tous ces blocs fonctionnels. Ainsi, il est possibles que certains paramètres ne soient pas utilisés pour un bloc fonctionnel donné.

2 Gmma : Graphe des modes de marche et d'arrêt

On propose de mettre en oeuvre une commande hiérarchisée suivant un GEMMA. Dans ce contexte : toute erreur de réseau (NOC_ETHIP_CONVOYEUR ou NOC_ETHIP_ROBOT) doit être vue comme une panne de la partie opérative et placer le système dans l'état D1. Le système reste dans cet état tant que la communication n'est pas rétablie. Pour chacun des mode, le sujet précise les blocs fonctionnels qui pourront être utilisés.

2.1 Mode D1 : Arrêt d'Urgence

Ce mode est le premier mode actif lorsque l'automate M340 passe en RUN. Dans ce mode :

- Le bras est mis hors puissance (MC_GroupPower) dès que cela est possible, c'est à dire :
 - La bibliothèque UnivalPlc est initialisée (T_StaubliRobot.Status.Initialized)
 - La baie CS9 est prête à recevoir des commandes (T_StaubliRobot.Status.Online)
- Le système de préhension est maintenu dans son état.
- Les bloqueurs du Poste Guichet doivent en interdire l'accès.

Pendant ce mode, la colonne lumineuse de l'IHM (hmiScrara.m_Outputs.m_qToR.m_nqx*Column) doit être :

- Rouge si le bras a pu être mis hors puissance,
- Rouge clignotant à la fréquence de 10 Hz sinon.

2.2 Mode D2 : Diagnostic/Traitement de la défaillance

Ce mode a pour objectif de donner une configuration saine du système. C'est à dire :

1. Acquitter les défauts présents (MC_GroupReset),
2. Vider la file des commandes de mouvements (MC_GroupStop),
3. Autoriser de nouvelles commandes de mouvement (MC_GroupContinue),
4. Mettre le bras sous puissance (MC_GroupPower).

Pendant ce mode, le guichet est interdit d'accès. La colonne lumineuse de l'IHM est au **jaune** fixe.

2.3 Mode A5 : Remise En Route après Défaillance

Ce mode permettra d'évacuer une pièce restée saisie. Nous développerons cette dépose dans un futur TP. Pour le moment, on considère que l'opérateur autorise la dépose par le biais du pupitre opérateur (bouton bleu « autorisation de dépose » – J212-2/J212-7 de la CS9) et signale que toutes les aires de traitement sont vides (bouton « cuves vides » de l'IHM). Le comportement sera donc le suivant :

1. Le système attend l'autorisation de dépose pour désactiver le système de préhension.
2. Le système attend que toutes les aires de traitement soient vides.

Pendant ce mode, le guichet est interdit d'accès. La colonne lumineuse de l'IHM est au **jaune** clignotant à la fréquence de 1 Hz.

2.4 Mode A6 : Mise En Conditions Initiales

On souhaite positionner le robot au-dessus du guichet, prêt à saisir un gobelet sans entraver la circulation des palettes vides ou pleines. Les coordonnées cartésiennes dans le repère « world » respectant ces contraintes sont : (X=90, Y=-410, Z=150, RX=-180, RY = 0, RZ =0).

Si la position courante n'est pas celle voulue, on adopte la stratégie suivante :

1. Monter l'axe J3 en position 150 mm sans modifier la position courante des autres axes.
2. Diriger le robot au-dessus du guichet pour terminer le positionnement voulu par une commande de mouvement de type **articulaire**.

Durant toute la durée de ce mode :

- Si un arrêt est demandé via le pupitre opérateur :
 - ◊ Immobiliser le robot tout en maintenant le bras sous puissance.
 - ◊ Reprendre la course dès que l'arrêt n'est plus actif.
 - ◊ Ce mode est signalé par l'allumage clignotant rapide (10 ms) du voyant rouge associé au bouton « arrêt demandé » ainsi que de la colonne lumineuse.
- Le guichet est interdit d'accès. Les deux bloqueurs (BE, BP) sont levés.

2.5 Mode A1 : Arrêt en conditions initiales

Le bras du robot sera immobile mais sous puissance au-dessus du guichet passant. La colonne sera *Cyan* afin d'indiquer à l'opérateur que la cellule est prête pour traiter des pièces en mode automatique s'il le souhaite.

2.6 Dans tous les modes

Le voyant blanc **Bras sous puissance** de l'IHM est allumé si et seulement si le bras est sous puissance.

3 Manipulations

3.1 Squelette du programme

Dans notre commande hiérarchisée, un programme principal `GM_Gmma` aura pour rôle de gérer les différents modes de marche du robot. C'est celui-ci qui autorisera ou non chaque mode de marche. Pour cela, on utilisera une section par mode de marche. Chacune des sections sera conditionnée à une variable booléenne commandée par le programme principal.

Le Gmma sera mis en oeuvre à l'aide de trois sections exécutés dans l'ordre suivant :

1. `GM_Gmma_Init`
 - Lors du premier cycle automate :
 - ◊ Initialisation du GMMA lors du premier cycle automate (à l'aide du bloc fonctionnel `InitChart`),
 - ◊ Configuration de la période du bit de vie du robot à 150 ms au sein de l'objet `staubliRobotScara`,
 - Mise en défaut du GMMA en cas d'erreur de communication.
2. `GM_Gmma_Main`
 - Gestion des différents modes de marche du robot
3. `GM_Gmma_Post`
 - Gestion des variables booléennes de chaque mode de marche,
 - Fournir l'état du GMMA à l'IHM (`hmiScara.m_Outputs.m_Screen.m_intStateGmma`).

Manipulation 1 : Préparation de la structure du programme

Dans un premier temps, nous allons créer le squelette de notre programme avec les différentes sections qui seront complétées tout au long de ce TP.

Étape 1 Créer les trois sections de gestion du GMMA.

Étape 2 Créer une section par mode de marche. Chaque section sera conditionnée à une variable booléenne correspondante. Clic droit -> Caractéristiques -> Condition

Manipulation 2 : Programmation du GMMA

Étape 3 Compléter la section `GM_Gmma_Init`

Étape 4 Compléter la section `GM_Gmma_Main`, les conditions pourront être écrites ultérieurement.

Étape 5 Compléter la section `GM_Gmma_Post`

3.2 Programmation des différents modes de marche

Pour chacun des modes de marches, on propose la démarche suivante :



Démarche

- Mode combinatoire :
 1. Réserver les éventuelles instances de blocs fonctionnels nécessaires
 2. Compléter la section correspondante en ST ou LADDER
- Mode séquentiel :
 1. Etablir le programme SFC de ce mode
 2. Créer les actions P1 pour configurer les mouvements à réaliser dans les étapes correspondantes
 3. Compléter la section d'appel des blocs fonctionnels associés
 4. Compléter la section *post* pour exécuter les différentes actions.

Manipulation 3 : Modes D1 et D2

Étape 6 Compléter le mode D1

Étape 7 Compléter le mode D2

Vérification 1 : Mode D1 et D2

Étape 8 Vérifier, compiler et téléverser le programme

Étape 9 Faire vérifier par l'enseignant

- ☐ Voyant blanc allumé sur le pendentif de la CS9
- ☐ Voyant blanc Bras sous puissance allumé sur l'IHM
- ☐ Guichet interdit d'accès
- ☐ Colonne lumineuse de l'IHM au jaune fixe pour le mode D2

Manipulation 4 : Modes A5

Étape 10 Compléter le mode A5

Vérification 2 : Mode A5

Étape 11 Vérifier, compiler et téléverser le programme

Étape 12 Faire vérifier par l'enseignant

- ☐ Guichet interdit d'accès
- ☐ Colonne lumineuse de l'IHM au jaune clignotant à la fréquence de 1 Hz

Manipulation 5 : Modes A6

Étape 13 Compléter le mode A6 en suivant la démarche proposée plus haut.

Vérification 3 : Mode A6

Étape 14 Faire vérifier par l'enseignant avant tout versement sur l'automate

Étape 15 Compiler, téléverser et vérifier le fonctionnement

Manipulation 6 : Mode A1

Étape 16 Compléter le mode A1 puis faire valider par l'enseignant