Logo

Description automatically generatedLogo

Description automatically generated

*Tutoriel :* ***USER***

Sources :

STAUBLI : https://www.staubli.com/global/en/robotics

**Mise en œuvre du**

**Robot STAUBLI**

**« Tx40  - CS8c »**

**/**

**« MCP-SRS »**

*2023/04/03*

Sommaire

[1 La société STÄUBLI 4](#_Toc131702147)

[2 Présentation d’une cellule robotique de formation (salle C204) 4](#_Toc131702148)

[3 Présentation des MATERIELS 5](#_Toc131702149)

[3.1 Système Robot : TX40 / CS8c 5](#_Toc131702150)

[3.2 Le Bras de robot 6 axes TX40 STÄUBLI : 6](#_Toc131702151)

[3.3 Contôleur CS8: 7](#_Toc131702152)

[3.4 MCP : boitier manuel 8](#_Toc131702153)

[3.4.1 Les touches du MCP : 8](#_Toc131702154)

[3.4.2 L’écran du MCP 8](#_Toc131702155)

[3.5 WMS : Working Mode Selector 9](#_Toc131702156)

[3.6 Chaine d’Arrêt d’Urgence du CS8c 9](#_Toc131702157)

[4 Mode de marche 9](#_Toc131702158)

[4.1 Comment se comporter avec le MCP 9](#_Toc131702159)

[4.2 Les modes de marche 10](#_Toc131702160)

[4.3 Procédure de mise en route de la cellule robotique : 10](#_Toc131702161)

[4.4 Déplacement au MCP 12](#_Toc131702162)

[4.5 BACKUP 13](#_Toc131702163)

[4.6 Libération des freins 13](#_Toc131702164)

[4.7 Commande Manuel des outils (Touches (12)) 13](#_Toc131702165)

[4.8 Déplacement du robot dans l’espace 13](#_Toc131702166)

[4.8.1 Position dans l’espace 13](#_Toc131702167)

[4.8.2 Orientation dans l’espace 14](#_Toc131702168)

[4.9 Déplacements du robot au boîtier manuel \_ MCP/CS8c 14](#_Toc131702169)

[4.9.1 Déplacement en mode articulaire (JOINT) : 14](#_Toc131702170)

[4.9.2 Déplacement en mode cartésien (FRAME, TOOL) : 14](#_Toc131702171)

[4.10 Gestionnaire d’application \_ MCP/CS8c 15](#_Toc131702172)

[4.10.1 Ouverture d’une application 15](#_Toc131702173)

[4.10.2 Fenêtre Application 16](#_Toc131702174)

[4.10.3 Structure d’une application 16](#_Toc131702175)

[4.10.4 Démarrage de l’Application 16](#_Toc131702176)

[4.10.5 Mouvement programme (MANUEL) 17](#_Toc131702177)

[4.10.6 Mouvement programme (AUTO LOCAL) 17](#_Toc131702178)

[4.10.7 Arrêt des mouvements : interruption mouvements programme 17](#_Toc131702179)

[4.10.8 En résumé : contrôle de l’application 17](#_Toc131702180)

[4.11 Sauvegarder et/ou Transférer votre Application 18](#_Toc131702181)

[4.11.1 Transfert du Robot / CS8c vers le PC 18](#_Toc131702182)

[4.11.2 Transfert du PC vers le Robot / CS8c 18](#_Toc131702183)

[4.12 APPRENTISSAGE DE LA CELLULE \_ MCP/CS8c 18](#_Toc131702184)

[4.12.1 Point articulaire : Variable de type joint 18](#_Toc131702185)

[4.12.2 Point cartésien : Variable de type point 19](#_Toc131702186)

[4.12.3 Configuration de Bras 19](#_Toc131702187)

[4.12.4 Sélection des outils 20](#_Toc131702188)

[4.12.5 Edition des outils 20](#_Toc131702189)

[4.12.6 Définition des outils 21](#_Toc131702190)

[4.12.7 Apprentissage des joints (articulaires) 21](#_Toc131702191)

[4.12.8 Apprentissage des points (cartésiens) 21](#_Toc131702192)

[4.12.9 Page d’apprentissage (MCP) 22](#_Toc131702193)

[4.13 Mode Point (vérification des points) \_ MCP/CS8c 22](#_Toc131702194)

[4.13.1 Ré-apprentissage : JOINT 22](#_Toc131702195)

[4.13.2 Ré-apprentissage : LINE 22](#_Toc131702196)

[4.13.3 Ré-apprentissage : APPRO 23](#_Toc131702197)

[4.13.4 Ré-apprentissage : ALIGN 23](#_Toc131702198)

[4.14 Mouvements programmés simples 23](#_Toc131702199)

[4.14.1 Mouvement Articulaire : movej 23](#_Toc131702200)

[4.14.2 Mouvement Linéaire : movel 24](#_Toc131702201)

[4.14.3 Mouvement Circulaire : movec 24](#_Toc131702202)

[4.14.4 Descripteur de mouvement mdesc (Acc,Vel,Dec,TVel,RVel,Blend (=lissage),Leave,Reach) 24](#_Toc131702203)

[4.14.5 Pile de Mouvements 25](#_Toc131702204)

[4.14.6 Synchronisation avec les mouvements : waitEndMove() 25](#_Toc131702205)

[4.15 Edition de programmes sur le MCP 26](#_Toc131702206)

[4.16 Frame définition 26](#_Toc131702207)

[4.16.1 Utilisation du Frame 26](#_Toc131702208)

[4.16.2 Apprentissage du Frame 26](#_Toc131702209)

[4.17 Exploitation du Frame 27](#_Toc131702210)

[4.17.1 Variable de type : trsf 27](#_Toc131702211)

[4.17.2 Palettisation dans un frame 27](#_Toc131702212)

[4.18 Approche sur points cartésiens 28](#_Toc131702213)

[4.18.1 Approche sur un point 28](#_Toc131702214)

[4.18.2 Approches : Exemple 1 28](#_Toc131702215)

[4.18.3 Approches : Exemple 2 28](#_Toc131702216)

[4.18.4 Approches : syntaxes possibles 29](#_Toc131702217)

[5 Programmation VAL3 29](#_Toc131702218)

[6 Exemples 30](#_Toc131702219)

[6.1 Exercice n°1 : Trajectoire avec descripteurs de mouvement, debogueur 30](#_Toc131702220)

[7 ANNEXES 34](#_Toc131702221)

[7.1 Société de robotique : 34](#_Toc131702222)

# La société STÄUBLI

**Robot STÄUBLI TX40**

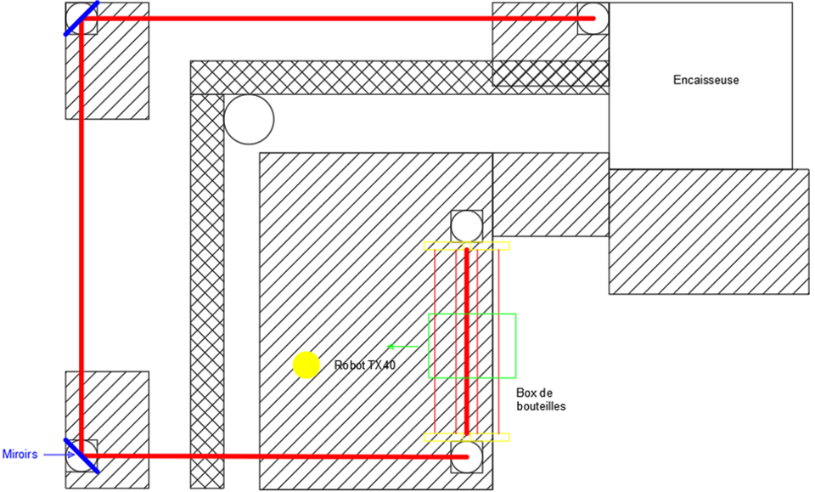
Création d’un compte sur le site internet de STAUBLI pour :

⮱ Avoir accès à une base de données : Programmer en Val 3 , Maj des soft…

# Présentation d’une cellule robotique de formation (salle C204)

A picture containing sky

Description automatically generatedA picture containing icon

Description automatically generated Vue 3D :

Une cellule est représentée par le schéma d’implantation ci-dessus. L’accès à une cellule se fait par une Barrière Immatérielle (BI). Le franchissement de la Barrière Immatérielle (BI) provoque l’arrêt du robot de la cellule.

Chaque poste est constitué de deux éléments :

La « Table Développeur », qui regroupe un PC, les commandes du robot utilisé (Boitier Manuel, sélecteur de mode de marche) ainsi qu’une dalle tactile qui gère les entrées /sorties ainsi que les acquittements de défaut du poste.

La « Table Robot » qui permet de réaliser les cycles robot pour les différents stages. L’accès à cette table est sécurisé en mode automatique par une BI.

**Sécurité des postes :**

Champ d’alarme de la « Table Robot » :

Une BI laser permet de protéger l’utilisateur lorsque le robot fonctionne en mode « Automatique ».

Toute intervention dans le champ d’alarme des BI provoque la coupure de puissance du bras concerné. Le défaut provoqué devra être acquitté via la dalle tactile pour autoriser à nouveau la mise sous puissance du bras.

L’intervention dans une cellule peut se faire via le franchissement des BI :

1- Eteindre le système « Encaisseuse » avec la CPU des BI (PnozMulti) en tournant le Bouton sectionneur rouge de l’encaisseuse sur OFF

2-

3- Une fois les barrières immatérielles alimentés, il faudra acquitter le « défaut BI » via la dalle tactile pour autoriser à nouveau la mise sous puissance du bras.

**Arrêt d’urgence d’une cellule :**

Tous les arrêts d’urgence présents sur la « Table Développeur » provoquent une coupure de puissance sur le bras du poste concerné.

Tout « Pendant d’apprentissage » connecté reste opérationnel (même éteint) au niveau de son arrêt d’urgence.

Les arrêts d’urgence « Cellule » et « BI » provoquent la coupure de puissance bras sur le poste de la cellule.

 Après un déverrouillage d’arrêt d’urgence, il faudra acquitter le défaut associé via la dalle tactile.

**Risques résiduels :**

Une analyse de risque a été réalisée pour définir les postes robots de la salle de Formation

Les dispositifs de sécurité en place permettent de maîtriser les risques identifiés

Comme dans toute installation, il reste toujours des risques résiduels, facilement maîtrisables en respectant quelques règles de base

Principalement, les distances de détection des BI sont inférieures à la préconisation de la norme TS 15066.

Il convient donc d’adopter une attitude responsable et prudente lors des manipulations sur robot, et ce pendant toute la durée de l’exercice :

**Ne jamais se précipiter vers un robot en fonctionnement**

**Marcher normalement** de manière à garantir la détection par les BI et l’arrêt des mouvements du robot

**Ne pas se positionner à l’intérieur de l’enceinte de la cellule**.

**Rester de préférence assis à la table développeur**, ou debout devant la table robot à une distance raisonnable, ne pas monter sur la table robot

**Rester vigilant et prudent à tout moment**, même lorsque le robot est en mode manuel

**Penser également aux personnes qui sont à proximité de votre robot lorsque vous l’utilisez, et ne pas laisser un robot en cycle sans surveillance**

# Présentation des MATERIELS

## Système Robot : TX40 / CS8c

Contrôleur

SP1 (option)

WMS (option)

A picture containing microscope

Description automatically generatedA picture containing electronics

Description automatically generated

A close-up of a computer

Description automatically generated with medium confidence

Bras TX

NB : « Option » WMS, système obligatoire mais pas forcément de conception STAUBLI (voir schéma de câblage

« Option » MCP, interchangeable avec un « Automate » et/ou le PC de développement avec le logiciel SRS

## Le Bras de robot 6 axes TX40 STÄUBLI :



**Préhenseurs** : - Ventouse → avec système « venturie »

* Pince Mécanique → fabrication « shunk »

⮱ Bride (flange) normalisée entre robot et préhenseur

⮱ 1 sortie électrovanne sur avant-bras pour le TX40

**Moteurs du robot**:

6 moteurs bruschless avec codeurs resolver

Codeur : information sur le positionnement

⮱ Codeur Resolver = codeur de type inductif (Codeur inductif : robustesse (pas de pièce mécanique mais un champ magnétique créé par des bobines) / Sensible au perturbation du au champ magnétique rapproché du moteur et du codeur / Supérieur à 4096 pt/tour ➔ changer de codeur.)

Les moteurs synchrones : pour un positionnement précis

⮱ **Brushless** (**av** : pas de ballais donc moteur moins encombrant moins d’usure pour la maintenance)

**Câblage du robot** : Contrôleur CS8 /**TX40**

⮱ se compose de 2 fibres optiques (FO) pour la transmission de données numériques issues des codeurs du ou des robots aux contrôleurs sur de longues distances

Les informations des 6 codeurs du robot sont filaires jusqu’à une carte électronique (à la base du robot), alimenté en 13V, qui transforment ces signaux pour la FO.

* **Positionner le Cache fibre optique sur le(s) connecteur(s) du câble lorsque celui-ci est déconnecté afin d’éviter les salissures**

**Câblage du contrôleur CS8c** :

De Base :

* User IO (~ms car CLKcs8=4ms) : Valve 1 et 2 + usrIn0 et usrIn1 entrées digitales (connecteur J109)
* Fast IO (20µs) : 2 entrées digitales et 1 sortie digitale
* ModBus TCP server (utilisation du robot en esclave seulement, sinon ajouter une carte Modbus client) : ⮱ connecteurs RJ45 sur les 2 premiers (Liaison assez lente ~100ms)

En Option :

* Carte BIO1 (16 E/S), Carte BIO2 (16 E/S), Carte Bus de terrain = CanOpen, ETHindustriel, …( visible dans le MCP : « tableau de bord ») outil de configuration : « AplicomIO »

**Connectiques robot TX40 : 2 positions**

Sélecteur pour le choix des freins des axes à libérer

Toujours le remettre à 0

* Connectiques « Horizontale » :



J1201 : Echappement

P3 : Entrée de pressurisation pour protéger les organes du robot de la saleté extérieur

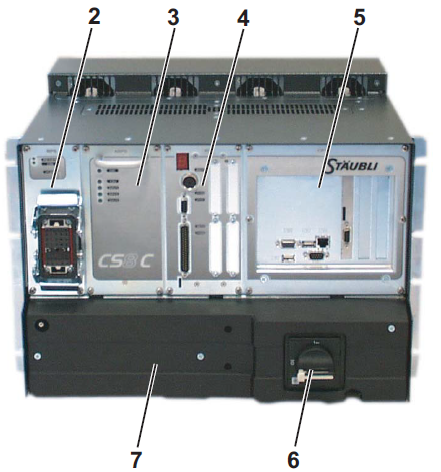
J1202 : Connecteur pour capteur (au niveau du préhenseur)

P2 : Alimentation pneumatique (directe), sortie P2 au niveau de l’avant-bras du robot

P1 : Alimentation électrovanne

Bouton pour valider la libération du frein moteur

* Connectique « Verticale » (sous le pied du robot) : Pour des cellules avec une nécessité de nettoyer le robot au « jet d’eau »)



## Contôleur CS8:

Amplificateurs

⮱ Avec OS temps réel : VxWorks

Connecteur du contrôleur :

Calculateur : 2,3,4,5

⮱ 10 ventilateurs

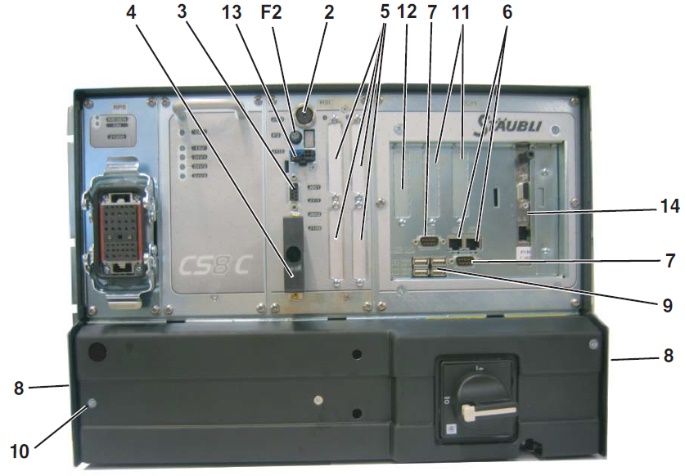
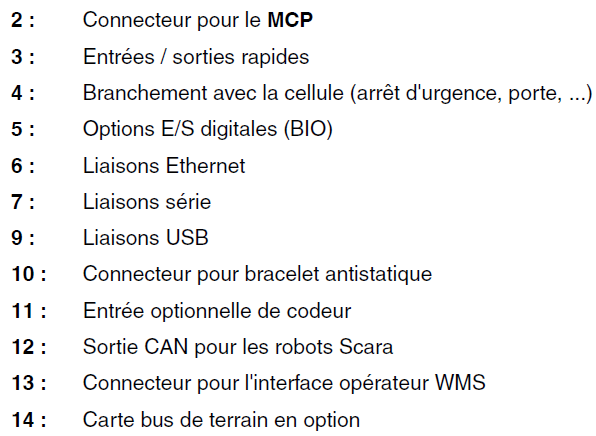
⮱ 32 entrées et 32 sorties

⮱ ModBus (gérer sur un connecteur RJ45)

Puissance : 6,7

⮱ Carte Ethernet Powerlink (option)

Asservissement : « écarte frein 24v » →voyant vert 24V2 sur le CS8 (libère les 6 freins du robot)



## MCP : boitier manuel

Il permet de :

⮱Visualiser des informations avec un Ecran

⮱Apprendre les points

⮱Programmer en Val3

⮱Maintenance

* Installer un « Connecteur schunt » lorsque celui-ci (MCP) n’est plus raccorder à la CPU : le contrôleur CS8 ici

### Les touches du MCP :

**Contrôle du Bras :**

1. Mode de marche :
2. Bouton de mise sous puissance du bras
3. Arrêt d’urgence
4. Touche de mouvement
5. Touche de choix du mode de déplacement : Joint / Frame /Tool / Point
6. Touche réglage de la vitesse

(13) Touche de jog

**Navigation dans interface :**

(7) Touches de menus contextuels :

(9) Touches d’interface et de navigation

**Saisie des données**

(8) Touches alphanumériques :

**Contrôle des applications**

(10) Touches de commande des applications :

**« HOMME PRESENT »** Bouton(11)

**Raccourcis sorties digitales**

(12) Touche d’activation de

### L’écran du MCP

[ **A** ] la barre d’état :

1. Indicateur de connexion au système et d’activité
2. Indicateur de présence de nouveaux messages d’information
3. Indicateur de saisies
4. Indicateur de statut du mouvement du bras : M / S / @ / .
5. Indicateur de vitesse de déplacement du bras
6. Indicateur de verrouillage des entrées et des sorties

[ **B** ] la plage de travail :

[**C** ] les menus contextuels :

## WMS : Working Mode Selector

Boitier avec clé 3 positions et bouton Arrêt d’Urgence (norme EN ISO 10218-1)

 Placer à l’**extérieur à la cellule**

3 positions avec la clé :

* Mode manuel
* Mode automatique
* Mode automatique déporté

Boitier à câbler sur le connecteur **J113** de RSI2 du CS8c

NB : A la place du kit WMS, l’intégrateur peu câbler son propre système

## Chaine d’Arrêt d’Urgence du CS8c

En mode manuel, les contacts AU de la chaîne automatique ne sont plus actifs

En opération manuelle, chaque personne présente à l’intérieur de l’enceinte doit disposer d’un équipement de protection de type « homme présent » (exemple : switch à l’arrière du MCP)

A picture containing graphical user interface

Description automatically generated

# Mode de marche

Mode Manuel : L’interface de mouvements manuels est un utilitaire dédié à la commande manuelle des mouvements du robot et à l’apprentissage des positions du robot. (p150/266 CS8c)

## Comment se comporter avec le MCP

Après interruption de production

* Lancer le premier cycle à faible vitesse (10%)
* Se tenir prêt à arrêter les mouvements du bras :

🡺 **Bouton d’Arrêt d’urgence (3)** : coupure générale

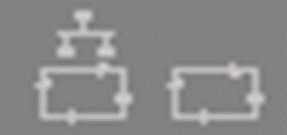
🡺 **Bouton de puissance (2)** : Coupure puissance Bras (Robot uniquement)

🡺 **Touche « Move/Hold »** : Arrêt soft et immédiat (Pas de coupure de puissance)

* **Suite à un arrêt d ’urgence en mode manuel, le pendant doit obligatoirement être placé sur son support pour permettre une remise sous puissance**

## Les modes de marche

**Mode Déporté**



**AUTOMATIQUE**: Mode de Production

* + La cellule est fermée, personne à l’intérieur

**Mode Local**

* + Le robot est contrôlé par un programme
  + Les mouvements peuvent être effectués à très grande vitesse



**MANUEL** : Apprentissage de points, Mise en position origine cycle

**Mode Lent** :

Joint, Frame, Tool

* Le robot est contrôlé par l ’opérateur qui a le boîtier manuel en main
* La vitesse est limitée à 250 mm/s maximum
* L’opérateur peut se tenir à proximité du robot

 La sélection doit se faire via un sélecteur (WMS) situé hors de la cellule robotisé (Non du pendant)

## Procédure de mise en route de la cellule robotique :

1. **AU :** Vérifier le bon fonctionnement de la chaîne d’arrêt d’urgence de la cellule robotique

( Sécurité par rapport à l’humain)

**🞏 Mise en service de la Barrière Immatérielle (BI) «**PILZ**»** du système « DEBOXEUR / ENCAISSAGE » pour un **Arrêt d’Urgence (AU) Système**

⮱ Alimenter la cellule robotique en 400V alternatif sinusoïdale : **Disjoncteur PC5**

⮱ Mettre sous tension le système « chaine d’embouteillage » avec le **sectionneur ON/OFF** sur **ON**

⮱ Lancer sur l’IHM B&R : **ARWin Start up** par un double touché sur l’écran / Attendre le menu vert

⮱ Lancer sur l’IHM B&R : **mappView** par un double touché sur l’écran :

Après l’identification : STEPHANE, vérifier la mise en place des BI avec la présence de **ligne verte** sur l’IHM du système. Dans le cas contraire, des **lignes rouges** nous informent d’une mauvaise orientation des BI ⇨ Manipulation sur BI avec les indicateurs visuels des BI PILZ.

⮱ Reseter les messages d’erreurs du système sur l’IHM avec les touches tactiles appropriées.

**🞏 Mise en service de la chaîne d’AU du robot**

⮱ Bouton arrêt d’urgence boîtier manuel **enclenché**

⮱ Bouton arrêt d’urgence cellule **enclenché** ou WMS

⮱ Sélecteur libération de freins tourné : mettre en position « 0 »

⮱ Fin de course sur une articulation ⇨ le positionner dans sa plage angulaire « soft » et non « électrique »

Graphical user interface, text, application

Description automatically generated

1. **CS8c : Mise sous tension du contrôleur du robot**

**🞏** Commuter le sectionneur général (6) en position ON

**🞏** Attendre l’apparition du menu principal du MCP

**🞏 Indicateurs visuels de fonctionnement :**

⮱ WMS : 1 LED sur 3 allumée avec la couleur verte fixe

⮱ MCP : LEDs clignotantes

⮱ Attendre que les leds s’éteigne ou arrête de clignoter

⮱ Ecran MCP : Menu Principal / Gestionnaire d’application (surligné)

⮱ CS8 : Attendre que l’afficheur 7 segments Affiche (Doc CS8 p226)

❑ « **c** » : arrêt d’urgence enclenché

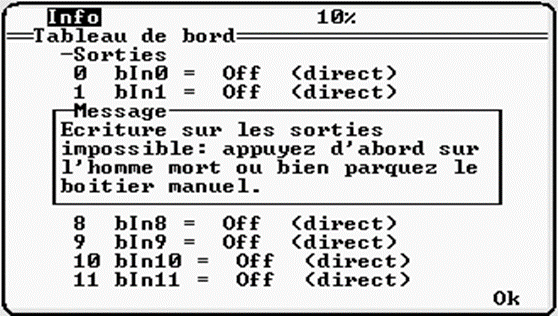
⮱ BI de la cellule robotique à Mettre en service (voir ci-dessous 4))

❑ « **U** » : Mémorisation AU, s’annule à la prochaine mise sous tension du bras

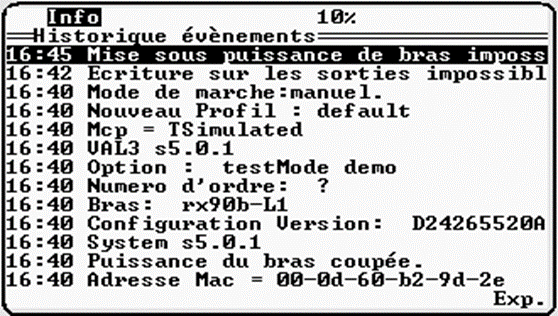
❑ «  **.** »  : **point clignotant = système OK et sous puissance**

❑ « **I .** » : Petit trait verticale et **point clignotant = système OK** sans la puissance ?

MCP : Messages d’erreurs 🡺 **Historique des évènements**

Quand « **Info** » clignote : Des messages n’ont pas été vus parce qu’un autre était déjà affiché

⮱ Aller à l’Historique d’évènements pour visualiser jusqu’à 250 messages (les plus récents) depuis dernier démarrage de la CS8

La touche **Export** permet de copier sur clé USB les fichiers d’évènement (fichiers principaux : errors.log, aussi possible : errors.old, assert.txt)

⮱ Fichier texte éditable sur un PC avec l’éditeur de votre choix (éviter le Bloc-Notes de Windows)

⮱ Messages avec la date et l’heure

1. **WMS** : Sélectionner le **mode manuel** avec le WMS

🞏 Pour sélectionner le mode manuel, tourner le sélecteur à clé à 3 positions dans la position appropriée. Le mode sélectionné est indiqué sur l’interface opérateur WMS et sur le MCP (repère 1)

1. **MCP :**

Ecran MCP : Ok -> menu principal

🞏 **Procédure MCP (MANUEL) de « Mise sous puissance du BRAS »**

Graphical user interface, application

Description automatically generated**➀** **Homme mort du MCP enfoncé**

**➁ Relâcher puis maintenir « homme mort » (11) dans sa position médiane au cours des 15 dernières secondes.** *(hint : Appui légé)*

**➂ Appuyer sur le bouton (2) de mise sous puissance du bras.**

⮱ Le voyant du bouton se met à clignoter pendant quelques secondes puis reste allumé fixe, cela signifie que le bras est desormais sous puissance et près à effectuer des mouvements.

NB : un nouvel appui sur le bouton (2) coupe la puissace sur le bras et active les freins. Il est alors nécessaire, pour rendre les mouvements possibles, de relancer la procédure de mise sous puissance du bras.

NB : En mode manuel, la puissance est également coupée si le bouton de validation est relâché ou si la position de la touche WMS a changé.

## Déplacement au MCP

Menu principal du MCP :

✓ **Gestionnaire d ’application** : Environnement programmation.

✓ **Gestionnaire de tâches** : Débug des applications en exécution.

✓ **Tableau de bord** : Affichage d ’informations, test des E/S.

✓ **Calibrage** : Recalage ou changement de moteur(s).

✓ **Historique évènements** : Historique des 250 derniers évènements depuis le démarrage du contrôleur.

NB : Gestion des Utilisateurs (default / Maintenance / Staubli) : Shift + User (sur le MCP) (p26)

NB : Navigation dans l’arborescence du MCP (voir doc constructeur) (p27)

Graphical user interface, application

Description automatically generated➀ **Vérifier** sur le MCP que vous êtes en mode manuel (attentionobligatoire !) et que le bras est sous tension

➁ **Sélectionner le mode de déplacement** :

⮱ Appuyer sur la touche **Joint** ou **Frame** Ou **Tool** ou **Point**

✓ Mode **joint** : déplacement axe par axe

✓ Mode **frame** (base) word : déplacement par rapport au sol

✓ Mode **tool** : déplacement par rapport à la pince

✓ Mode **point** : apprentissage

⮱ le voyant correspondant (5) s’allume.

Icon

Description automatically generated Déclarer **Au moins un point articulaire (mode joint) dans un pg**

 Outil : flange = Bride (pas de pince)

 En mode manuel, la vitesse de déplacement est limitée à 250 mm/s

 La vitesse indiquer en % sur le MCP (mode manuel) est beaucoup plus lente que la vitesse déclarer dans le programme pour un fonctionnement en mode automatique

⮱ **Ne pas oublier de remettre la vitesse du MCP à une valeur entre 25 et 50%**

➂ **Bouger les axes** en Appuyant sur l’une des touches de déplacement :

**J1** (axe1)

**J2** (axe2)

**J3** (axe3)

**J4** (axe4)

**J5** (axe5)

**J6** (axe6)

**(4) : jog**

**ou** **(13) : mini jog**

**ou** en mode Point sur la touche « **Move/Hold** »



 **Ne pas aligner les axes 3/4/5/6 pour un mvt : Problème de singularité (=pb avec les modèles mathématiques) :vrai pour tous robots**

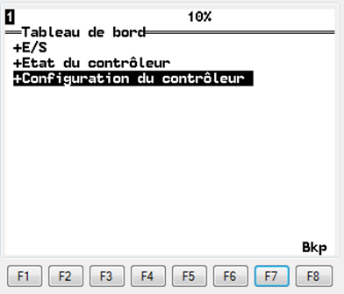
Arrêt / reprise des mouvements avec la touche STOP : touche Stop puis touche Run

Arrêt des mouvements avec le Bouton d’arrêt d’urgence : ne pas utiliser pour un simple arrêt

## BACKUP

Quand effectuer un backup de votre contrôleur CS8c :

* A la livraison
* A chaque maintenance
* A la suite d’une mise à jour importante
* A la suite de modifications importantes

**Procédure de Backup :**

* Insérer une clé USB dans le CS8c (connecteur J202 ou J209)
* Touche « Menu » du MCP
* Sélectionner le « **Tableau de bord** » avec la touche « Enter » du MCP
* Touche « 🡺 » sur la ligne « Configuration du contrôleur
* Touche « F8 » du MCP : **Bkp**

Backup complet du système : 3 partitions = **USR, SYS, LOG**

## Libération des freins

Pourquoi : déplacer un axe manuellement

Procédure :

* Être en mode manuel et hors puissance
* Sélecteur de frein : choisir l’axe à déplacer
* Appuyer sur le Bouton « Libération » et déplacer l’axe sélectionné
* **Remettre le sélecteur de frein sur 0 sinon impossible de remettre la puissance.**



## Diagram Description automatically generatedCommande Manuel des outils (Touches (12))

Possibilité de réaffecter les touches à d’autres sorties

**Icon

Description automatically generated** **1** électrovanne seulement pour le TX 40

## Déplacement du robot dans l’espace

### Position dans l’espace

Par défaut nous sommes dans un repère **WORLD** qui se situe au niveau de l’épaule du robot

Diagram

Description automatically generated

 Sens de l’axe X, à l’opposé de la plaque signalétique et de raccordement électrique du robot (ou suivre la marque sur la fonderie du robot si connectique verticale)

NB : **X+** Pouce de la main droite

NB : **Y+** Index de la main droite

NB : **Z+** vers le haut = doigt vers le haut (majeur)

A picture containing linedrawing

Description automatically generated

{**X,Y,Z**} : Coordonnées de la **position** du robot =

Décalages le long des trois axes pour passer du 0,0,0 à la position cartésienne du CDO (=TCP) robot

### Diagram Description automatically generatedOrientation dans l’espace

Règle du point de la main droite avec le pouce comme axe de référence et les doigts donnent le sens de rotation autour de l’axe Rn+, Rn- (ou règle du tir bouchon)

Diagram

Description automatically generated

{**RX,RY,RZ**} : Coordonnées de l’**orientation** du robot

=

Rotations autour des trois axes pour passer du 0,0,0 à l’orientation du CDO robot

## Déplacements du robot au boîtier manuel \_ MCP/CS8c

### Déplacement en mode articulaire (JOINT) :

Les touches (4) ou (13) permettent d’effectuer des mouvements en mode articulaire « **joint** » autour des différents axes. Ces mouvements se fonts dans le sens + ou -.

****Procédure de déplacement :

* Mode : **Manuel**
* Touche **Joint**
* Réglage vitesse **+/-**
* Touches **+/-** de X,Y,Z,RX,RY,RZ correspondant respectivement aux joints J1 à J6

### Déplacement en mode cartésien (FRAME, TOOL) :

#### Repère (FRAME) = Repère de base fixe : « world »

Diagram

Description automatically generated

Les mouvements peuvent être effectués en translation et en rotation

⮱**En translation** **: X / Y / Z** : règle des 3 doigts de la main droite (Pouce : X / Index : Y / Majeur : Z)

⮱ **En rotation** : **RX / RY / RZ** (Règle du point de la main droite avec le pouce comme axe de référence et les doigts donne le sens de rotation autour de l’axe Rn+, Rn-)

**Axes de déplacement** : axes **parallèles** à ceux du repère **frame** **fixe** (**WORLD** par défaut) passant par l’**origine** du **repère outil** courant (TCP)

Procédure de déplacement :

* **A picture containing icon

  Description automatically generated**Mode : **Manuel**
* Touche **Frame**

**NB** : Max de l’enveloppe = max de l’amplitude du robot = sur la ligne X : si l’outil est sur la ligne X = plus grande sera son amplitude

* Réglage vitesse **+/-**
* Touches **+/-** de X,Y,Z,RX,RY,RZ (le CDO ou TCP se déplace suivant les axes du repère WORLD)

#### A close-up of a helmet Description automatically generated with low confidenceRepère (TOOL) = Repère outil mobile : « tool »

Les mouvements peuvent être effectués en translation et en rotation

⮱**En translation** **: X / Y / Z** : règle des 3 doigts de la main droite (Pouce : X / Index : Y / Majeur : Z)

⮱ **En rotation** : **RX / RY / RZ** (Règle du point de la main droite avec le pouce comme axe de référence et les doigts donne le sens de rotation autour de l’axe Rn+, Rn-)

**Axes de déplacement** : axes du **repère outil** mobile courant (**FLANGE** par défaut) ( : Axe **Z+ sort de l’outil**)

A close-up of a camera

Description automatically generated with low confidence

 : Axe **Z+ sort de la bridel**)

****Procédure de déplacement :

* Mode : **Manuel**
* Touche **Tool**
* Réglage vitesse **+/-**
* Touches **+/-** de X,Y,Z,RX,RY,RZ (le CDO ou TCP se déplace suivant les axes du repère Flange(tool)

Si la touche « **Tool**» a été actionnée, les déplacements se font parallèlement aux axes de l’outil courant

(Flange par défaut)

#### Déplacement en mode Point (POINT) :

⮱ **Utile pour vérifier la position d’un point appris dans l’espace**

## Gestionnaire d’application \_ MCP/CS8c

### Ouverture d’une application

Procédure « ouverture application » :

* Insérer une clé USB dans le CS8c (connecteur J202 ou J209)
* Touche « Menu » du MCP
* Sélectionner le « **Gestionnaire d’application** » avec la touche « Enter » du MCP
* Touche « **Ouv** »
* Text

  Description automatically generated**C:\Users\jdul\Desktop\Next Button Icons Vector Pack\PNGs\White On Black\next-button-9.png**Touche « 🡺 » sur la ligne Disque ou **USB0**
* Aller sur la ligne de l’application souhaité avec ou
* Valider par « **Ok** »

NB : Ouverture d’une application : Manuel / Ouv.auto / Exec.auto (p41)

### Fenêtre Application

Diagram

Description automatically generated

**Exp.** : pour le versioning de nos programmes

**Rech.** : il faut recharger l’application après un transfert

### Structure d’une application

* VARIABLES GLOBALES : variables de différents types définis par le programmeur :

⮱ Outils, repères, points cartésiens, positions angulaires,

⮱ Booléens, numériques, chaînes de caractères

* PROGRAMMES : programmes (exécutés comme tâche ou sous-programme).

A picture containing graphical user interface

Description automatically generated

* Edition de données :

⮱Pour éditer un champ (Entrer la valeur ou la sélectionner dans une liste)

⮱ Pour valider

### Démarrage de l’Application

Procédure de démarrage de l’application :

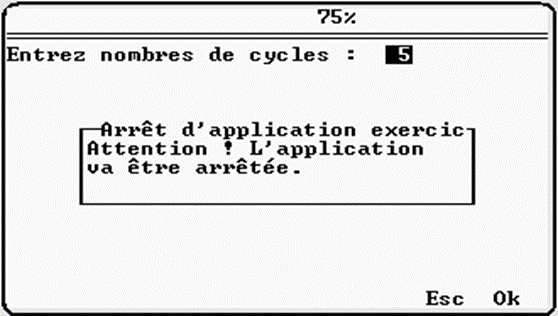
* Mise sous puissance avec le MCP
* …
* Valider l’application sélectionner : **Ok**

Démarrer l’application en appuyant sur la touche « **Run »**

⮱ Application en exécution : **led orange** à côté de la touche « run »

* Donner l’ordre de mouvement en maintenant enfoncé

la touche **Move/Hold**

Pour arrêter une application « proprement »,

* Appuyer sur la touche **STOP**
* Valider l’arrêt sur le MCP : **Ok**

### Mouvement programme (MANUEL)

### Diagram Description automatically generatedMouvement programme (AUTO LOCAL)

### Arrêt des mouvements : interruption mouvements programme

À tout moment, il est possible d ’interrompre les mouvements programme

* **Touche MOVE/HOLD** : Arrêt / reprise des mouvements

⮱ En mode manuel, les mouvements du bras sont activés lorsque la touche Move/Hold est pressée. Dès que la touche est relâchée, le bras s’arrête immédiatement sur la trajectoire programmée.

⮱ Le système mémorise alors un point d’arrêt

⮱ Pour commander le Mouvement de reprise en mode local ou manuel, appuyer sur la touche Move/Hold :

* **Touche STOP**

⮱ Arrêt / reprise des mouvements avec la touche STOP / puis touche Run

* **Bouton puissance ou AU** : coupe la puissance (ne pas utiliser pour un simple arrêt)
* Changer de mode de marche pour poursuivre dans un autre mode.
* Déplacer le robot au boîtier manuel en mode LENT
* Déplacer le robot en libération de frein

**Sans interrompre l’exécution de l’application**

### En résumé : contrôle de l’application

Diagram

Description automatically generated

NB : A chaque coupure du contrôleur la vitesse moniteur est conservé

## Sauvegarder et/ou Transférer votre Application

### Transfert du Robot / CS8c vers le PC

ROBOT (**MCP**)

* Arrêter l’application (Run puis Stop avec MCP)
* Enregistrer / Fermer / Exporter (versionnage) avec MCP
* **Transfert manager** sur SRS
* Recharger (rafraichir) avec SRS

PC (**SRS**)

### Transfert du PC vers le Robot / CS8c

PC (**SRS**)

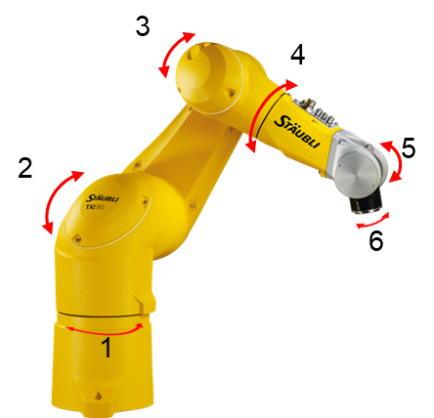
* Enregistrer (-sous) les modifications avec SRS
* **Transfert Manager** sur SRS
* Recharger (après un transfert depuis SRS) avec le MCP

ROBOT (**MCP**)

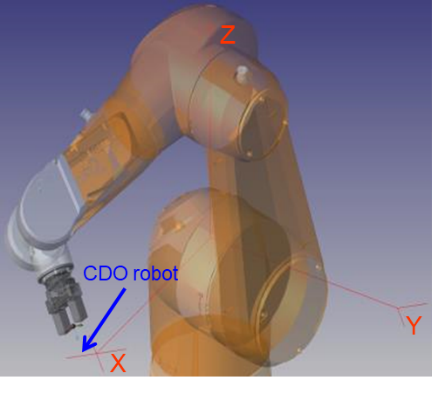
## APPRENTISSAGE DE LA CELLULE \_ MCP/CS8c

* : **Pour apprendre un point, IL FAUT AVOIR DEFINI UN OUTIL !**

### Point articulaire : Variable de type joint

* ****Variables dans lesquelles une position du robot est stockée sous la forme de **6 (6 axes) valeurs** correspondantes aux positions de chaque articulation du bras.
* Ces 6 valeurs représentent les **positions en degrés** de chacune **des 6 articulations** du bras.
* La position zéro de chaque articulation est matérialisée par un repère physique (marques de fonderie).
* L’intérêt de ces **points articulaires joint** est qu’ils **imposent** de manière complètement **déterministe** la **posture du bras.**
* Leur limitation est le fait qu’ils **ne peuvent pas être utilisés** dans tous les types de **trajectoire**.
* **Ces points articulaires joint sont donc utilisés comme des points de configuration, dans l’espace ouvert, en approche des différentes zones de travail du robot.**

### Point cartésien : Variable de type point

* Variables dans lesquelles une **position du robot** est stockée sous la forme de **6 coordonnées :**

**3 translations X, Y, Z** / **3 rotations RX, RY, RZ**

* Un **point cartésien** est toujours relatif à un **référentiel**, qui par défaut est le **repère world**, mais qui peut aussi être un repère **frame utilisateur** (un point cartésien en VAL3 contient cette information en son sein).
* Les **6 coordonnées** d’un **point cartésien** représentent la **position et l’orientation du CDO** (Centre De l’Outil) exprimées par rapport au **référentiel** du point. (CDO = TCP).
* Un **point cartésien** contient également des informations de **posture du bras** (moins déterministe qu’un joint), qui sont par défaut réglées pour **interdire au robot de changer sa posture** lors d’un mouvement vers ce point. Elles sont exprimées sous la forme :
  + D’une configuration d’épaule
  + D’une configuration de coude
  + D’une configuration de poignet
* **L’intérêt des points cartésiens est qu’ils peuvent être utilisés comme destinations de mouvements en ligne droite ou en arc de cercle**

### Configuration de Bras

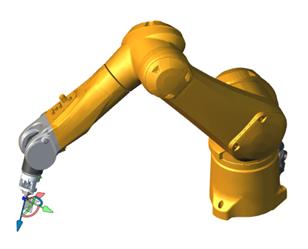
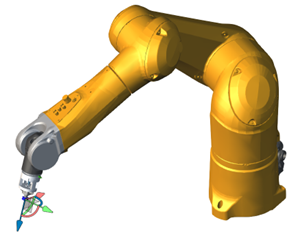
**Epaule et Coude :**

Epaule Droite : **RIGHTY** 🡺 **Bras à droite de l’épaule** (connecteur dans le « dos », sinon repère fonderie)

Epaule Gauche : **LEFTY** 🡺 **Bras à gauche de l’épaule**

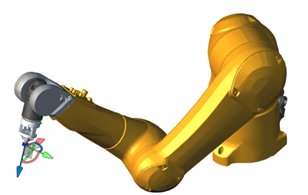
**NEGATIVE**

**POSITIVE**

**RIGHTY**

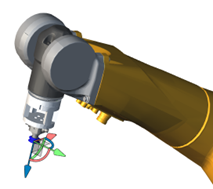
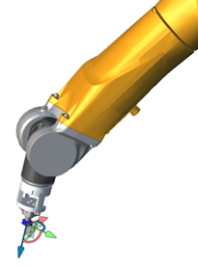
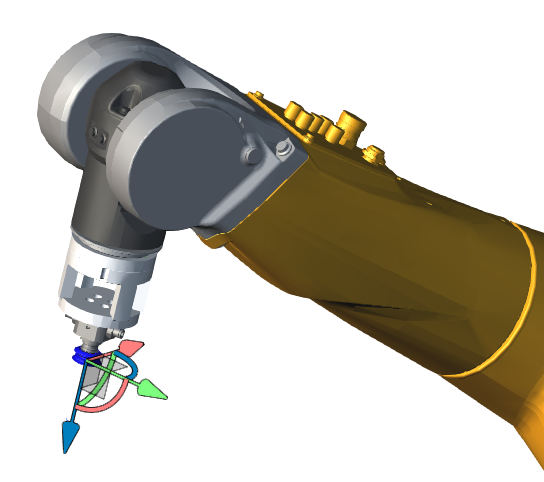
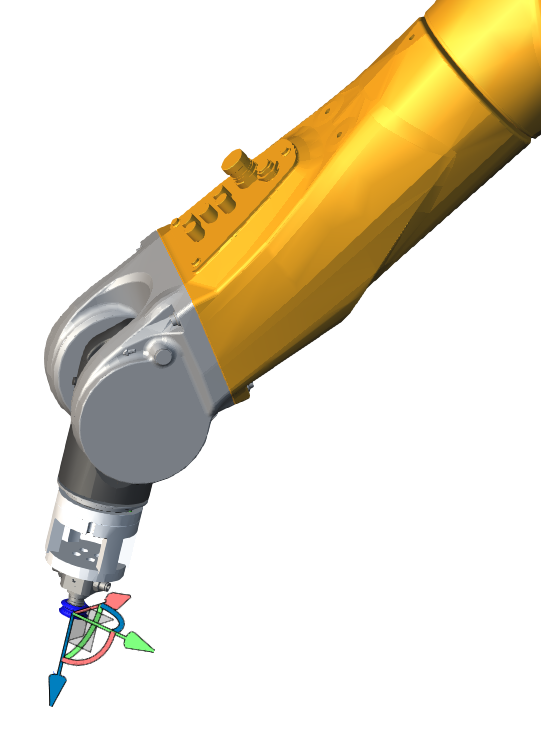
**LEFTY**



**NEGATIVE**

**LEFTY**

**Poignet**: Configuration positive si les connecteurs de l’avant-bras sont vers le haut



**NEGATIVE**

**POSITIVE**

### Sélection des outils

Text

Description automatically generated**Lors de manipulation de points cartésiens, il faut impérativement sélectionner l’outil de travail**

Text

Description automatically generatedA picture containing table

Description automatically generated

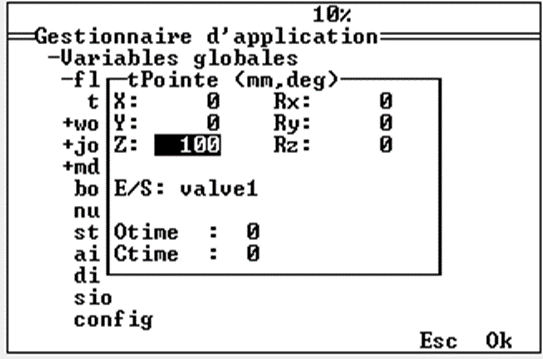
**1**

### Edition des outils

**Procédure :**

**MCP / Gestionnaire d’application  / Application VAL3 / +exnn / +Variables globales**

**/ +Flange : tPince /OK**

**Tableau**

**X=**0 **RX=**0

**Y=**0 **RY=**0

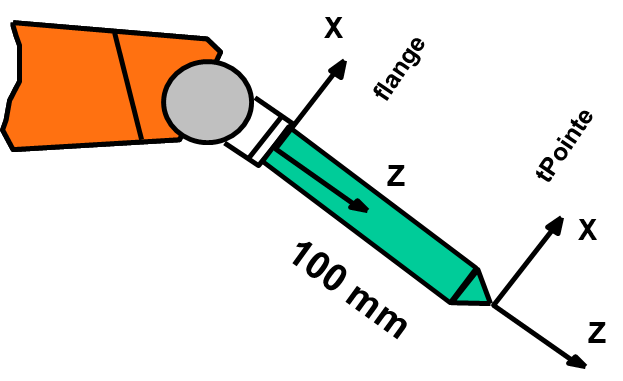
**Z=**180**mm** **RZ=**0

* **Z=180 🡺 lors des tests, le robot exécutera les mouvements appris avec un recul de 180mm**

**E/S :** valve**1** : pour l’ouverture ou la fermeture de la pince

**Otime= 0** : tps d’ouverture de pince

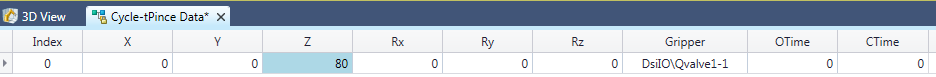
**Ctime = 0** : tps de fermeture de pince



**Intérêt pour :**

* **Apprentissage des points**
* **Contrôle vitesse et position**
* **Correction géométrique (en cas d’usure)**

### Définition des outils



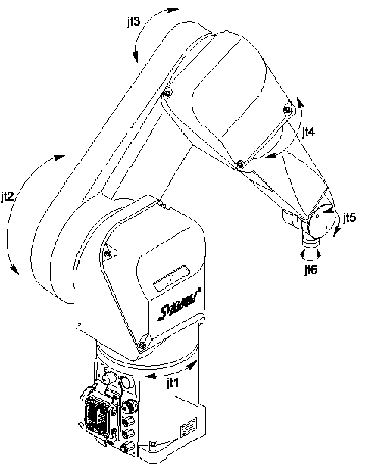
Diagram

Description automatically generated

**Définir la géométrie et l’action d’un outil, permet de :**

* **Atteindre les mêmes points avec différents outils**
* **Contrôler la vitesse et la trajectoire en bout d’outil**
* **Corriger la géométrie de l’outil au fil du temps (en cas d’usure d’un outil porté par le robot)**
* **Faciliter l’apprentissage des points**
* **Les coordonnées d’un point correspondent à la position d’un centre outil à un moment donné (apprentissage)**

### A picture containing text Description automatically generatedApprentissage des joints (articulaires)

A picture containing text

Description automatically generatedGraphical user interface, application

Description automatically generated

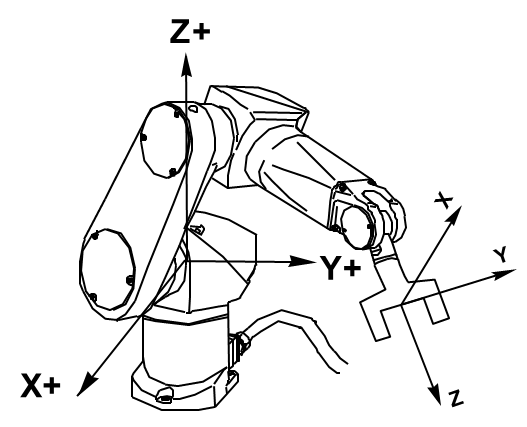
Utiliser les différentsmodes de déplacements

### Apprentissage des points (cartésiens)

Diagram

Description automatically generated

Utiliser les différentsmodes de déplacements

Text

Description automatically generatedGraphical user interface, application

Description automatically generated

### Diagram Description automatically generatedPage d’apprentissage (MCP)



## A picture containing text Description automatically generatedMode Point (vérification des points) \_ MCP/CS8c

### Ré-apprentissage : JOINT

⮱ Vérifier ou réapprendre des points en mode JOINT

A picture containing text, receipt, screenshot

Description automatically generated

1. Clic sur **Mode**

⮱ Type mouvement = Mode : **JOINT**

1. Clic sur **Par.**

Appro = distance d’approche par rapport au point

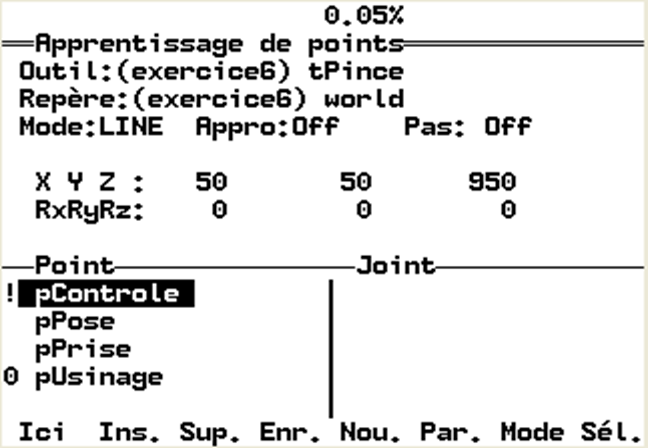
⮱ Avec ou sans Appro : sélection de la direction + distance

1. Clic **pPose**
2. Clic **Move/Hold**: Contrôle du mouvement

NB : Sél.=Sélection outil

### Ré-apprentissage : LINE

⮱ Vérifier ou réapprendre des points en mode LINE



1. Clic sur **Mode**

⮱ Type mouvement = Mode : **LINE**

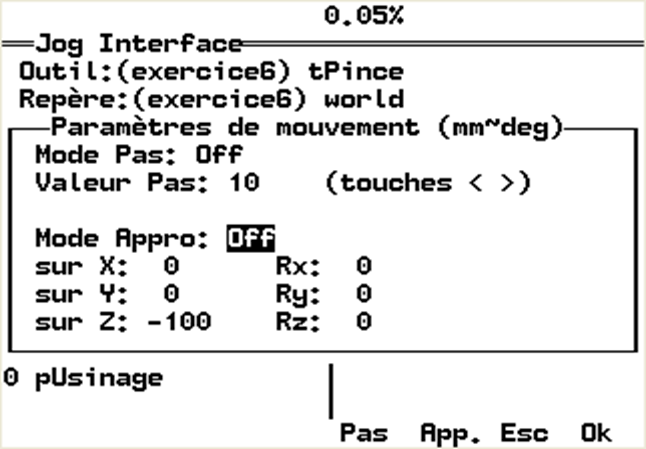
1. Clic sur **Par.**

Appro = distance d’approche par rapport au point

⮱ Avec ou sans Appro : sélection de la direction + distance

1. Clic **pControle**
2. Clic **Move/Hold**: Contrôle du mouvement

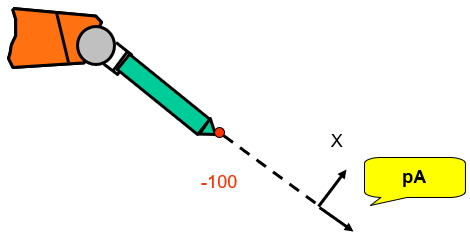
### Ré-apprentissage : APPRO

⮱ Vérifier ou réapprendre des points : APPRO

1. Clic sur **App.**

⮱ Type mouvement = Mode Appro : **Off/On**

1. Clic sur **Z : -100**
2. Clic **Move/Hold**: Contrôle du mouvement



### A picture containing diagram Description automatically generatedRé-apprentissage : ALIGN

⮱ Vérifier ou réapprendre des points en mode ALIGN

**Aligner l’axe Z de l’outil courant sur l’axe le plus proche du repère courant**

1. Clic sur **Mode**

⮱ Type mouvement = Mode : **ALIGN**

1. Repère**: world**
2. Outil : **tPince**
3. Clic **Move/Hold**: Contrôle du mouvement

NB : Mode ALIGN par rapport à frame

⮱ Axe Z tool // Axe Z frame

## Mouvements programmés simples

### Mouvement Articulaire : movej

* **Mouvement courbe**
* **Vitesse et accélération décrite par mdesc (descripteur de mouvements)**

Syntaxe:

**movej(point,tool,mdesc)**

ou

**movej(joint,tool,mdesc)**

movej(pA,tPince,mRapide)

movej(pB,tPince,mRapide)

movej(pC,tPince,mRapide)

movej(pD,tPince,mRapide)

movej(pE,tPince,mRapide)

### Chart, line chart Description automatically generatedMouvement Linéaire : movel

* **Mouvement en ligne droite**
* **Vitesse et accélération décrite par mdesc (descripteur de mouvement)**
* **Mouvement utilisé pour insérer des pièces . . .**

Syntaxe:

**movel(point,tool,mdesc)**

NB : non disponible pour le type joint

movel(pA,tPince,mRapide)

movel(pB,tPince,mRapide)

movel(pC,tPince,mRapide)

movel(pD,tPince,mRapide)

movel(pE,tPince,mRapide)

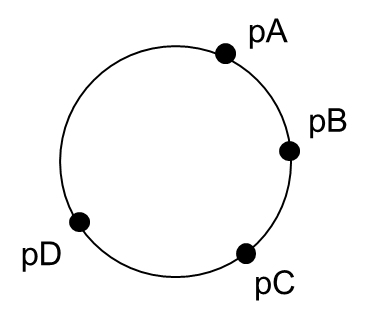
### A picture containing shape Description automatically generatedMouvement Circulaire : movec

* **Interpolation circulaire : mouvement en arc de cercle**

Syntaxe :

**movec(point,point,tool,mdesc)**

**! Deux points**

****NB : non disponible pour le type joint

NB : **Cercle réalisable avec 4 points :**

**movec(pB,pC,tPince,mRapide)**

**movec(pD,pA,tPince,mRapide)**

movel(pA,tPince,mRapide)

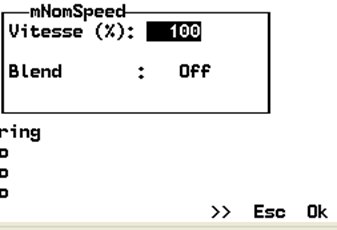
**movec(pB,pC,tPince,mRapide)**

movel(pD,tPince,mRapide)

### Descripteur de mouvement mdesc (Acc,Vel,Dec,TVel,RVel,Blend (=lissage),Leave,Reach)

Graphical user interface, application

Description automatically generated

****

* **vel, accel, decel** : **%** des valeurs nominales sur articulations

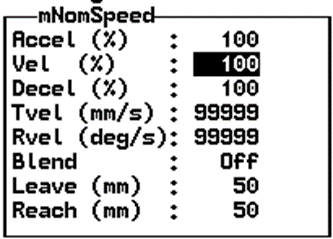
NB : 100% par rapport à la charge donnée par la doc

⮱ Ex : VEL=110% et ACC/DEC=120% 🡺 valeur max si la charge

est inférieure à la valeur nominale.

* En complément : pour un contrôle type process :
* **tvel** : vitesse de translation max. en bout outil en mm/s,
* **rvel** : vitesse de rotation max. en bout outil en degrés/s

Hors process : on garde les valeurs définies

* Au final : Valeur la plus restrictive sera utilisée



* **Blend : lissage de la trajectoire, du plus rapide au plus lent :**

**- joint** : la trajectoire du CDO(1) n’est plus contrainte entre les points leave et reach.

- **Cartesian** : le CDO(1) peut s’écarter de la trajectoire entre les points leave et reach, mais doit rester dans le plan de cette dernière

- **off** : pas de lissage, le CDO(1) s’arrête sur tous les points

1. **CDO** : Centre De l’Outil (=**TCP**)

Text

Description automatically generated

### Graphical user interface, text Description automatically generatedPile de Mouvements

Graphical user interface, text

Description automatically generatedDiagram

Description automatically generated

**Bouton**

**Mouvement/Pause**

**!!! COMPLETE INDEPENDANCE ENTRE LES 2 NIVEAUX !!!**

**Sur certaines applications, il faut stopper et relancer une application pour prendre en compte les dernières coordonnées apprises**

### Synchronisation avec les mouvements : waitEndMove()

* **waitEndMove()** permet de bloquer l’exécution du programme jusqu’à ce que le robot arrive à un point spécifique

⮱ Le robot ira alors jusqu’au point cible du dernier mouvement à exécuter, et s’y arrêtera (le lissage est annulé)

Chart, line chart

Description automatically generated

**ACCEL VEL DECEL TVEL RVEL BLEND LEAVE REACH**

Exemple : mRapide = {100, 100, 100, 99999, 99999, joint, 20, 20}

movel(pA,tPince,mRapide)

movel(pB,tPince,mRapide)

movel(pC,tPince,mRapide)

**waitEndMove()**

movel(pD,tPince,mRapide)

**De ce fait, le waitEndMove() doit s’utiliser aussi peu que possible, et uniquement quand le robot doit vraiment aller jusqu’au point !**

NB : TVEL et RVEL =99999 🡺 pas de limitation en vitesse ici

NB : Lissage utile ici pour effectuer des déplacements plus rapide

## Edition de programmes sur le MCP

Voir doc « Formation Programmeur

## Frame définition

Pourquoi un frame ?

Mon robot est en production, l’application tourne à plein régime mais ….Une mauvaise manipulation et ...!!!! CATASTROPHE !!!!! …une journée d’apprentissage de point.. sauf si ...

### Diagram Description automatically generatedUtilisation du Frame

Repère de travail local :

* **Pour faciliter le ré apprentissage des points**
* **Utiliser pour dupliquer des points**
* **Décalage de points dans palette**

A picture containing diagram

Description automatically generated

⮱ Assistant d’**App**rentissage

### Apprentissage du Frame

Procédure :

MCP : Gestionnaire d’application / Variables globales / world / **App.**

/ Définir 3 points à apprendre du nouveau Frame « **fRef1** »: **Point Origine**, **Point X**, **Point Y**

/ Utiliser un outil précis et défini comme « courant » (ex : **tPointe**)

/ Apprendre les points les plus écartés possible (+ de précision) avec la touche **Ici** du MCP

NB : X et Y le plus loin possible de l’origine

Shape, arrow

Description automatically generated

Diagram

Description automatically generatedText

Description automatically generated

## Exploitation du Frame

### Variable de type : trsf

* Les variables transformées cartésiennes de type TRSF s’utilisent dans des opérations mathématiques sur les points cartésiens :

⮱ Exemple : **Approche sur point, décalage dans une palette , ….**

* Elles se composent de 6 champs numériques : x, y, z, rx, ry, rz
* Écritures possibles dans un programme *:* ***trsf******trDecal*** déclarée comme une variable globale ou locale :

⮱ Initialisation de tous les champs d’un coup : ***trDecal*** *=****{0,0,-100,0,0,0}***

⮱ Initialisation champ par champ :

trDecal.x=0 trDecal.rx=0

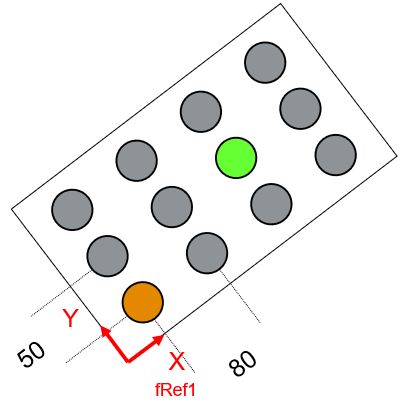
trDecal.y=0 trDecal.ry=0

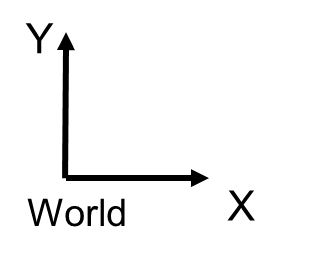
trDecal.z=-100 trDecal.rz=0

* Impossible de faire des mouvements sur une TRSF : utilisable uniquement dans des calculs

### Palettisation dans un frame

***Compose(point,frame,trsf)*** : **calcule un point décalé de trsf suivant le repère frame**



Code :

**p6=compose(pA,fRef1,{160,50,0,0,0,0})**

**movel(p6 ,tPince, mLent)**

p6

pA

NB :

* **compose** 🡺 calcul avec **frame**
* **approche** 🡺 calcul avec **tool**

## Approche sur points cartésiens

### Approche sur un point

* Permet de calculer par programme un point d’approche
* Les décalages sont appliqués par rapport aux axes de l’outil (ou aux axes propres du point, ce qui revient à dire la même chose)
* Avantage majeur : moins de points appris, les points d’approche et de dégagement sur les posages sont calculés par rapport aux points finaux de prise/dépose

 **A ne pas confondre avec compose() qui permet de faire des décalages par rapport aux axes d’un repère FRAME**

Code : **appro(point, trsf)**

### Approches : Exemple 1

Code :

1. movej(**appro(pPrise,{180,0,-100,0,0,0})**,tPince,mLent)
2. movel(**appro(pPrise,{0,0,-100,0,0,0})**,tPince,mLent)
3. Diagram

   Description automatically generatedmovel(**pPrise**,tPince,mLent))

### Approches : Exemple 2

Code :

1. movel(pPrise,tPince,mLent)
2. movel(**appro(pPrise,{0,0,0,0,-20,0})**,tPince,mLent)

Diagram

Description automatically generated(Blend =off)

### Approches : syntaxes possibles

* Soient les variables : point **pApp ;** point **pPrise ;** trsf **trDecalz ;** num **nDistance=100**
* Mouvement vers le point d’approche :
* Ecriture décomposée (plus de variables, plus de lignes, mais moins longues) :

Diagram

Description automatically generated

⮱ **trDecalz={0,0,-nDistance,0,0,0}**

⮱ **pApp=appro(pPrise,trDecalz)**

⮱ **movej(pApp,tPince,mRapide)**

* + Imbrication des instructions (moins de variables, ligne plus longue)

⮱ Pour pouvoir modifier simplement les distances/axes d’approche :

**movej(appro(pPrise,trDecalz),tPince,mRapide)**

⮱ Pour pouvoir modifier simplement **uniquement** la distance d’approche :

**movej(appro(pPrise,{0,0,-nDistance,0,0,0}),tPince,mRapide)**

⮱ Pour « empêcher » la modification simple des distances/axes d’approche :

**movej(appro(pPrise,{0,0,-100,0,0,0}),tPince,mRapide)**

# Programmation VAL3

Voir documentation STAUBLI « Formation Programmeur »

# Exemples

## Exercice n°1 : Trajectoire avec descripteurs de mouvement, debogueur

On demande :

Ecrire un programme VAL3 pour réaliser une trajectoire avec des segments en lignes droites

Utiliser les supports de stylos pour réaliser la trajectoire sur une feuille (voir pièce jointe)

1. Créer et renseigner l’outil « tStylo »
2. Créer et apprendre le joint « jDepart »
3. Créer et apprendre les points pA à pJ
4. Ecrire le programme de mouvement réalisant la trajectoire en segments de droites
5. Paramétrer un lissage sur toute la trajectoire exceptée au point pC et pD.

Icon

Description automatically generated Après chaque modification relative au mouvement, toujours exécuter le programme en mode manuel d’abord, puis en mode automatique

Solution Ex1 :

Procédure : Apprentissage des points

1. **Mise sous tension chaine d’embouteillage** : encaissage avec robot cartésien et les 2 convoyeurs

* Mettre sous tension le système « chaine d’embouteillage » avec le **sectionneur ON/OFF** sur **ON**
* Lancer sur l’IHM B&R : « **ARWin Startup »** par un double touché sur l’écran : Attendre le menu vert ARwin=**RUN**
* Lancer sur l’IHM B&R : « **MappView »** par un double touché sur l’écran

Après l’identification : STEPHANE, vérifier les BI : présence de **lignes vertes** sur le pourtour de la cellule robotique puis **reseter les messages d’erreurs** du système signalés d’une croix rouge (Prise d’origine, BI, …) : Appui touche sur le message d’erreur suivi d’un appui touche sur le bouton « reset » représenté par *Icon

Description automatically generated✓(Autant de fois que de messages)*

1. **Mise en air** :

⮱ Vérifier sur le mano dédié au robot cartésien « **6** bars » minimum dans l’armoire et la poignée bleue sur ON

⮱ Vérifier sur le mano dédié au robot STAUBLI « **6** bars » minimum et le bouton rotatif rouge sur ON

1. **Mise sous puissance du CS8 et du Bras TX40**

➀ **CS8**: commuter le sectionneur général (l’écran du MCP s’allume) :

* Attente écran stable avec apparition du « menu principal » à l’écran.
* Attendre que l’afficheur 7 segments Affiche (Doc CS8 p226)

❑ « **c** » : arrêt d’urgence enclenché

❑ « **U** » : Mémorisation AU, s’annule à la prochaine mise sous tension du bras

❑ « **.** »  : **point clignotant = système OK**

➁ **WMS** (clé) : mode manuel -> *indiqué par une Led verte fixe*

➂ **AU** du MCP et du WMS libéré

➃ **Bras TX40** : sélecteur frein en « 0 »

➄ **MCP** Homme mort enfoncé dans sa position médiane

➅Appuyer sur le bouton vert

1. **MCP** :

Menu Principale :

**Gestionnaire d’application**

**/ 🡪** *(pour valider le chemin)*

**/ Application VAL3**

**/** **F8 : nouv**

**/** **Nom** : « **ex1x** » (pour nouveau) …

ou **/ F7 : ouv** (ou **🡪**)

**/ +Disque**

**/ 🡪**

**/-Disque** « nom de l’application » si déjà enregistré

**/** **Nom** : choisir le nom de l’application que nous souhaitons ouvrir = « **ex1x** »

**/ 🡪**

**/ +ex11** (surligné)

(**/ Esc :** *pour remonter dans le menu si besoin)*

**/ 🡪**

**/ +Variables globales**

**/ 🡪**

Icon

Description automatically generated ***pour éditer un champ taper «⮠*** ***»*** puis valider ***«⮠*** ***»***

**/ +Flange** : tStylo /OK

**Tableau**

**X=**0 **RX=**0

**Y=**0 **RY=**0

**Z=**180**mm** **RZ=**0

* **Z=180 🡺 lors des tests, le robot exécutera les mouvements appris avec un recul de 180mm**

**E/S :** valve**1** : pour l’ouverture ou la fermeture de la pince

**Otime= 0** : tps d’ouverture de pince

**Ctime = 0** : tps de fermeture de pince

**/ F8 : OK**

⮱ Icon

Description automatically generated Plusieurs méthodes pour enregistrer un point :

**/ Nov** Nom : **point** (nom du point) **/ OK / OK**

Effectuer un mouvement pour aller au point de coordonnées choisi en mode joint ou frame

**/ Ici** (F1) : mémorise la position du point

/ **Ok** (F8) 🢡 **@** **point** sur le MCP

1. Apprentissage des points :

**/ +Joint** : pour enregistrer le point de départ

**Au moins un point articulaire (joint) dans un pg**



Icon

Description automatically generated *pour taper une valeur* ***«⮠*** ***»***

**/ 🡪**

Nom = j**depart**

**/ OK / tableau / OK**

ou **/ 🡪 si déjà enregistré**

*Effectuer un mvt sur le point choisi avec la touche du mode joint :*

**/ Move :** (F1)

**/ Ici** (F1) : mémorise la position du point

**/ Ok (F8)** 🢡 **@** j**depart** sur le MCP

**/ Esc :** *pour remonter dans le menu si besoin et choisir le menu World)*

(ou **/ Sél / Ok (F8) / Mode : (F7)**)

**/ +World** (frame) : points sur la feuille

Icon

Description automatically generated Conseil : apprentissage des points avec la pièce dans la pince

**/ 🡪**

Nom = pp**a / OK / OK**

à

pp**g** **/ OK / OK**

*Effectuer un mvt sur le point choisi avec la touche mode frame*

**/ Ici** (F1) : mémorise la position du point

/ **Ok** (F8) 🢡 **@** p**px** sur le MCP

**/ Esc :** *pour remonter dans le menu si besoin et choisir le menu* ***mdesc****)*

1. Créer variable mLent :

**/ +mdesc** (mLent=30% / blend=**off**)

**/** **Nou**: Nlle variable **/ OK** ou **/ 🡪** si déjà enregistré

**/ mlent / OK**

Vitesse =  **↵** **30** % **/ OK**

Blend = ↵ **Off** **/ OK**

**/ Esc :** *pour remonter dans le menu si besoin et choisir le menu* ***Programme****)*

1. Editer le programme :

**/ +Programme**

**/ +ex1x** **/ Edit** (edition d’un programme) : **F4**

**begin**

//remise en position attente

* ***pour entrer une nouvelle valeur (a,2,..)***

*Editer un champ «***⮠**  *» puis valider celui-ci «***⮠** *»*

* *Move****j :*** *mvt courbe*
* *Mov****l :*** *mvt ligne droite*

move**j**(jdepart, tstylo,mlent)

movej(ppa, tStylo,mlent)

move**l**(ppb, tStylo,mlent)

movel(ppc, tStylo,mlent)

movel(ppd, tStylo,mlent)

movel(ppe, tStylo,mlent)

movel(ppf, tStylo,mlent)

movel(ppg, tStylo,mlent)

movel(pph, tStylo,mlent)

movel(ppi, tStylo,mlent)

movel(ppj, tStylo,mlent)

movel(ppa, tStylo,mlent)

movej(jdepart, tStylo,mlent)

**waitEndMove()** Icon

Description automatically generatedcar la scrutation est trop rapide ⇨ le pg ne se fait pas

**end**

(F8) : Enr.

**/ Esc :** *pour remonter dans le menu si besoin et choisir le menu* ***Start****)*

Icon

Description automatically generated **Mettre ex1x dans start :**

**/ +Start / Edit** (edition d’un programme) : **F4**

**begin**

call **ex1x()** (Icon

Description automatically generated au parentheses)

**end**

1. **Exécuter le programme en mode manuel :**

Graphical user interface, company name

Description automatically generated

✓ Démarrer l’application en appuyant sur la touche **Run** du MCP

⮱ Ecran MCP : Lancement d’application

Application Val3 / Ouv / Disque

ex1x (surligné)

Ok (F8)

⮱ Indicateur Visuel MCP :

Graphical user interface, application, chat or text message

Description automatically generated Touche RUN : led orange Fixe

Move Hold : led bleue clignotante

✓ Donner l’ordre de mouvement en maintenant enfoncé la touche **Move/Hold**

⮱ En mode manuel, les mouvements du bras sont activés lorsque la touche Move/Hold est pressée. Dès que la touche est relâchée, le bras s’arrête immédiatement sur la trajectoire programmée.

Le système mémorise alors un point d’arrêt

Pour commander le Mouvement de reprise en mode local ou manuel, appuyer sur la touche Move/Hold

✓ Arrêt des mouvements avec la touche **STOP**  /Ok /Ok

⮱ Reprise des mouvements avec la touche Run

✓ Arrêt des mouvements avec le Bouton d’arrêt d’urgence : Icon

Description automatically generatedne pas utiliser pour un simple arrêt

✓ Ouverture et fermeture de la pince avec un appui touche sur le bouton 1 du MCP

Icon

Description automatically generated Tourner la valve rouge pour alimenter en air le robot SATUBLI

Le 1 avril 2022 exercice : ex12

Utiliser un crayon laser pour l exercice

Procédure réapprentissage des points :

1. Outil
2. Toul / frame
3. Ici
4. Test
5. **modif**

Voir pour lisser les points

En mode automatique

# ANNEXES

## Société de robotique :

**Robot system** : 44 Thouaré sur loire

**Sepro** : robots Cartésiens (4 axes) à la roche/yon

**Sea productique**

**U2 robotics:** ABB, … **(**Vendée**)**