Logo

Description automatically generated Logo

Description automatically generated



*Tutoriel :*

Sources :

PILZ

BR : WEB *« br-automation.com »*

AS *« Help »*

Tutoriels : *« Démarrer 1 axe avec mapp Axis »*

*/ « cnc avec mapp Axis »*

**Mise en œuvre d’un « Déboxeur » sur une chaine d’embouteillage**

**Avec un robot 6 axes STAUBLI et une CPU B&R**



MS 2023/07/06

**Table des matières :**

[I. B&R / STAUBLI / COGNEX / PILZ 4](#_Toc139625549)

[1) Le système : 4](#_Toc139625550)

[2) Synoptique du système : 4](#_Toc139625551)

[II. Présentation des matériels 5](#_Toc139625552)

[1. PILZ : 5](#_Toc139625553)

[1) PILZ : Sécurité Machine (Voir rapport : « Sécurité machine ») 5](#_Toc139625554)

[2) PILZ : Logicielles « PNOZmulti configurator » 5](#_Toc139625555)

[3) PILZ : Matériels (Voir rapport : « Tuto\_Pilz\_DeboxEncais ») 5](#_Toc139625556)

[4) PILZ : Disposition 5](#_Toc139625557)

[2. B&R : Robot cartésien 3 axes + Convoyeurs 2 axes 6](#_Toc139625558)

[1) Logicielles : 6](#_Toc139625559)

[2) Matériels : 6](#_Toc139625560)

[3) Câblage des Matériels : 6](#_Toc139625561)

[3. B&R : IHM robot Staubli (avant de merger sur la CPU « APC910 ») 6](#_Toc139625562)

[4) Logicielles : 6](#_Toc139625563)

[5) Matériels : 6](#_Toc139625564)

[4. COGNEX : Caméra de vision 6](#_Toc139625565)

[5. STAUBLI : Contôleur CS8c et Bras robotique 6 axes TX40 6](#_Toc139625566)

[III. STAUBLI / DEBOXEUR: Robot 6 axes 7](#_Toc139625567)

[1. STAUBLI / DEBOXEUR : Synoptique : 7](#_Toc139625568)

[2. STAUBLI Logiciel : 7](#_Toc139625569)

[3. STAUBLI Matériels : 8](#_Toc139625570)

[1) Liste des matériels 8](#_Toc139625571)

[2) Le bras de robot TX40 8](#_Toc139625572)

[3) Connectiques robot TX40 : 8](#_Toc139625573)

[4) Contrôleur CS8 9](#_Toc139625574)

[5) WMS : Boitier 9](#_Toc139625575)

[6) MCP : Pupitre de commande 9](#_Toc139625576)

[7) STAUBLI Moteurs des bras du robot 10](#_Toc139625577)

[4. STAUBLI : Câblage des Matériels 10](#_Toc139625578)

[5. STAUBLI : Modes de Marche 11](#_Toc139625579)

[1) Procédure de mise en énergie de la cellule robotique : 11](#_Toc139625580)

[2) Mode Manuel 12](#_Toc139625581)

[6. Relier B&R à STAUBLI 🡺 .xdd : pour une liaison EPL 13](#_Toc139625582)

[1) Comment trouver le fichier .xdd ? 13](#_Toc139625583)

[2) Importer le fichier .xdd dans la CPU B&R 15](#_Toc139625584)

[3) Configuration du module importé par le fichier xdd : Com B&R / STAUBLI 16](#_Toc139625585)

[IV. STAUBLI Programmation 19](#_Toc139625586)

[1. Logiciel de développement SRS 2022 : 19](#_Toc139625587)

[2. Le programme VAL3 avec SRS 19](#_Toc139625588)

[3. La communication POWERLINK 20](#_Toc139625589)

[4. Sauvegarde des données avec une cellule STAUBLI : SRS ↔ CS8c 20](#_Toc139625590)

[5. Programmer avec SRS : Procédure exemple avec le déplacement du robot sur 6 points 21](#_Toc139625591)

[1) Contexte 21](#_Toc139625592)

[2) Procédure Exemple avec le programme : PgDebox6pts 21](#_Toc139625593)

[6. Programmation Déboxeur 27](#_Toc139625594)

[1) Problématique : 27](#_Toc139625595)

[2) Les variables utilisées dans les programmes AS et SRS 27](#_Toc139625596)

[3) Créer les touches de start et stop dans AS 27](#_Toc139625597)

[7. Maintenance : 28](#_Toc139625598)

[V. Annexes 29](#_Toc139625599)

[1. Structure Réseau « DEBOXEUR » 29](#_Toc139625600)

[2. SRS 2022 29](#_Toc139625601)

[3. STAUBLI / Carte POWERLINK 30](#_Toc139625602)

[1) Indicateurs visuels : 30](#_Toc139625603)

[2) Nœud Carte STAUBLI / POWERLINK 31](#_Toc139625604)

[4. Programmation exemple avec le projet « Deboxeur17 » 32](#_Toc139625605)

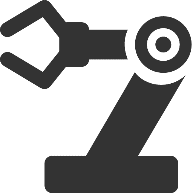
# B&R / STAUBLI / COGNEX / PILZ

### Le système :

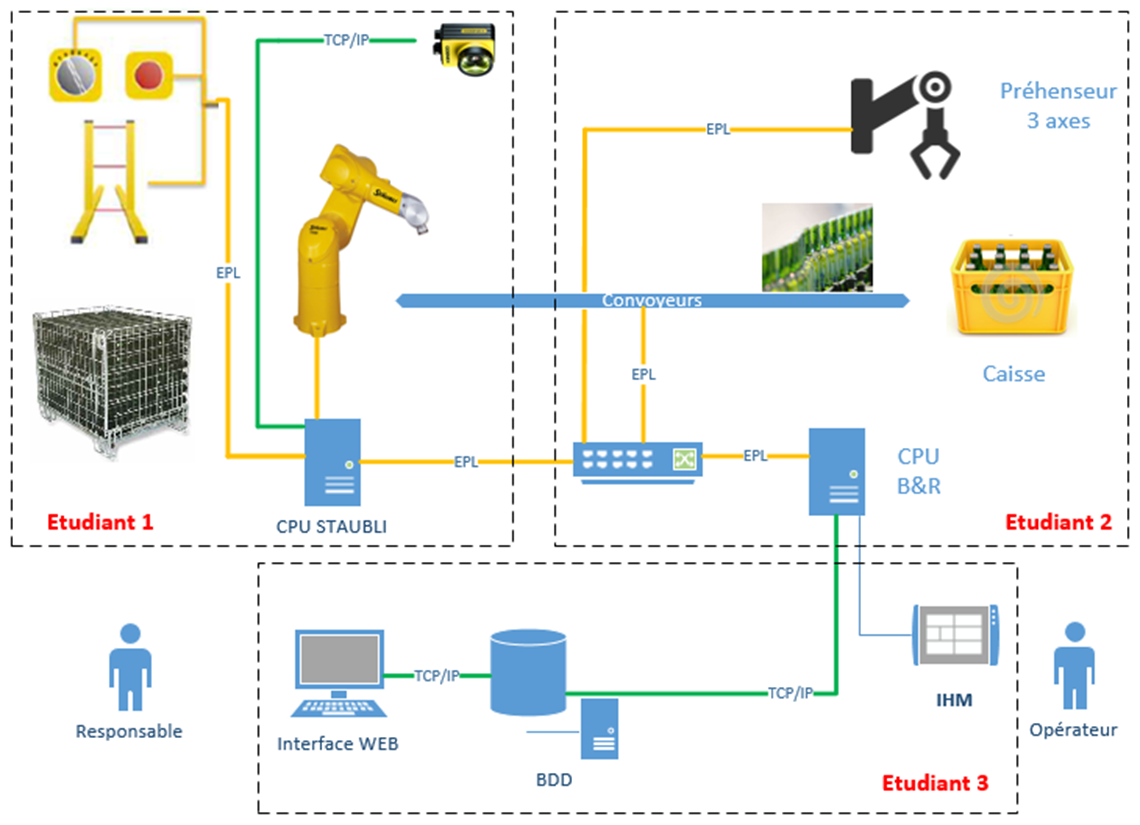


Logo

Description automatically generated



### Synoptique du système :



# Présentation des matériels

## PILZ :

### PILZ : Sécurité Machine (Voir rapport : « Sécurité machine »)

### PILZ : Logicielles « PNOZmulti configurator »

### PILZ : Matériels (Voir rapport : « Tuto\_Pilz\_DeboxEncais »)

* **CPU** : PNOZmulti
* **AU** :
* **BI**:  **BI**de chez PILZ : ref

**BI Muting** de chez PILZ : ref

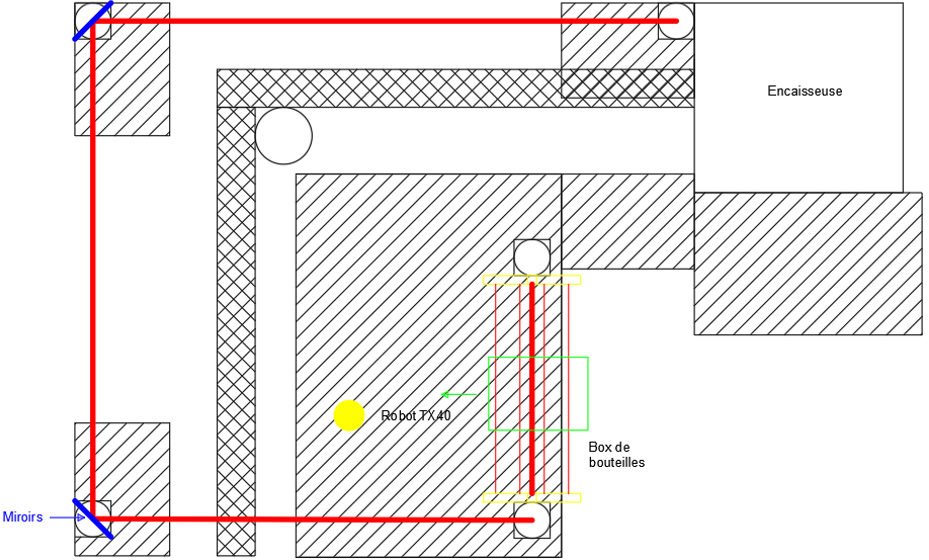
**Porte Muting**de chez PILZ : ref

* **Com** **Port EPL** avecle **n° de nœud = 05** *(voir roue codeuse)*

**USB** :

* **Mécanique** : Support BI

### PILZ : Disposition



## B&R : Robot cartésien 3 axes + Convoyeurs 2 axes

### Logicielles :

AS **AS4.6.**

AR AREmbended (**AR** **B4.53** ici)

### Matériels :

* **CPU** : **APC910** / com : EPL

5PP520.1043.00

5PP5CP\_US158-02

5PP5IF.FPLM-00

* **Ecran :**
* **AU** : **AU**  (x3)
* **Safety**:  **BI**(PILZ)**:**
* **Com** **Port EPL**

X20 BC 0083 (Bus Contrôleur EPL avec le **n° de** **nœud : 02**)

**ETH**: **@IP** apc910 : **10.16.7.30 (Win) / 10.16.7.31 (ARwin)**

* **Drive Variateur** : ACOPOS Inverter (400V) pour les 2 axes des convoyeurs

ACOPOS Alone 1010 (400V) pour le moteur synchrone

X20 pour le moteur à courant continu

**Moteur** : ***NB***: on ne les configure pas, ils seront découverts plus tard avec AS

🡺 Synchrone (brushless) 1 8LSA35 EA030D000-0 (3000 tr/mn)

🡺 Courant Continu

**Pneumatique : 🡺** Vérin

* **Mécanique** : **Robot Cartésien 3 axes**

**Pince pilotée par air**

### Câblage des Matériels :

(cf Annexes)

## B&R : IHM robot Staubli (avant de merger sur la CPU « APC910 »)

### A screenshot of a computer Description automatically generatedLogicielles :

AS **AS4.6.3.55**

AR AREmbended (**AR** **B4.62** ici)

### Matériels :

* **CPU** : **PPc70** / com : EPL
* **Ecran : 7 pouces**
* **Com** **EPL** avec **nœud : 02**

**ETH**avec **@IP** =**10.16.7.34**

## COGNEX : Caméra de vision

Caméra Insight 7000

* **Com** : **ETH** avec @IP= **10.16.7.19**

**EIP** (EtherNet/IP)

## STAUBLI : Contôleur CS8c et Bras robotique 6 axes TX40

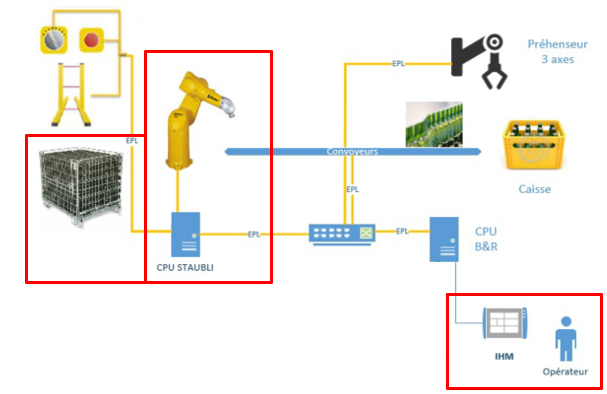
Aller au chapitre III ci-dessous

# STAUBLI / DEBOXEUR: Robot 6 axes

## STAUBLI / DEBOXEUR : Synoptique :

La partie robotique comprend trois éléments (voir annexe « *Rapport Robotique* ») :

* Le robot TX40 de Staübli
* L’ordinateur de contrôle CS8C de Staübli
* L’ordinateur APC 910 et son écran (panel) B&R



## STAUBLI Logiciel :

SRS 2022

## STAUBLI Matériels :

### Liste des matériels

* **Robot :** **TX40** / 6 axes
* **CPU** : Contrôleur STAUBLI : **CS8C**
* **Pupitre de Commande** (ou pendant d’apprentissage) :

**MCP**

* **Com** **EPL : carte POWERLINK** avec le **nœud = 3** (voir roue codeuse)

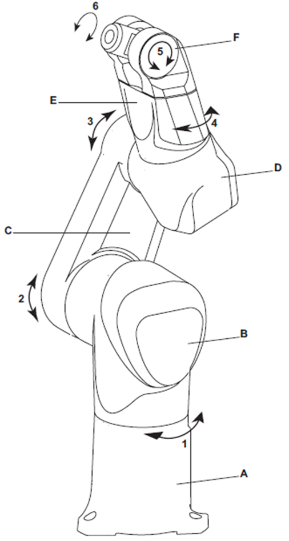
**ETH**: @IP cs8: **10.16.7.18**

* **Drive Moteur** : (voir annexe)
* **Mécanique** : **BRAS** Robot 6 axes STAUBLI**: TX40**
* **Pneumatique : Préhenseur type Pince mécanique pilotée par air**

- Ventouse → entreprise « venturie »

* Pince Mécanique → entreprise « shunk »

⮱ Bride (flange) normalisée entre robot et préhenseur



### Le bras de robot TX40

**F : Poignet**

**E : Avant-Bras**

**D : Coude**

**C : Bras**

**B : Epaule**

**A : Pied**

attention **NE PAS ALIGNER LES AXES 3/4/5/6 POUR UN MVT**

⮱ Pb avec les modèles mathématiques

(Vrai pour tous les robots 6 axes)

### Connectiques robot TX40 :

Sélecteur pour le choix des freins des axes à libérer

Toujours le remettre à 0, sinon impossible de remettre la puissance



J1201 : Echappement

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

Bouton pour valider la libération du frein moteur

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

P1 : Alimentation électrovanne

J1202 : Connecteur pour capteur (au niveau du préhenseur)

NB : sous tension, possibilité de libérer les freins

### Contrôleur CS8

⮱ Avec OS temps réel : VxWorks

Connecteur du contrôleur :

⮱ 10 ventilateurs

⮱ 32 entrées et 32 sorties

⮱ ModBus (gérer sur un connecteur RJ45)

⮱ Carte Ethernet Powerlink (option)

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

### WMS : Boitier

A picture containing microscope

Description automatically generatedBoitier avec clé 3 positions et bouton Arrêt d’Urgence

( **extérieur à la cellule**)

3 positions avec la clé :

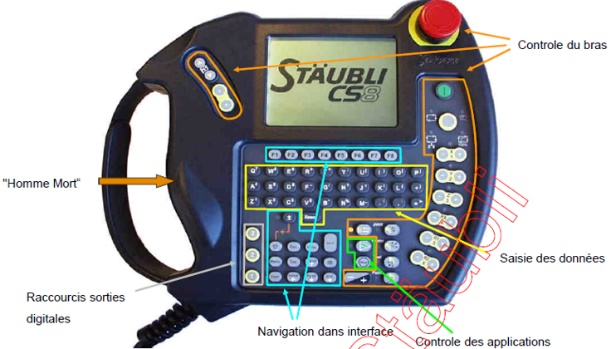
* Mode manuel
* Mode automatique
* Mode automatique déporté

### MCP : Pupitre de commande

**Présentation du MCP**

Il permet de :

⮱Visualiser des informations avec un Ecran monochrome (14 lignes / 40 caractères)

⮱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*

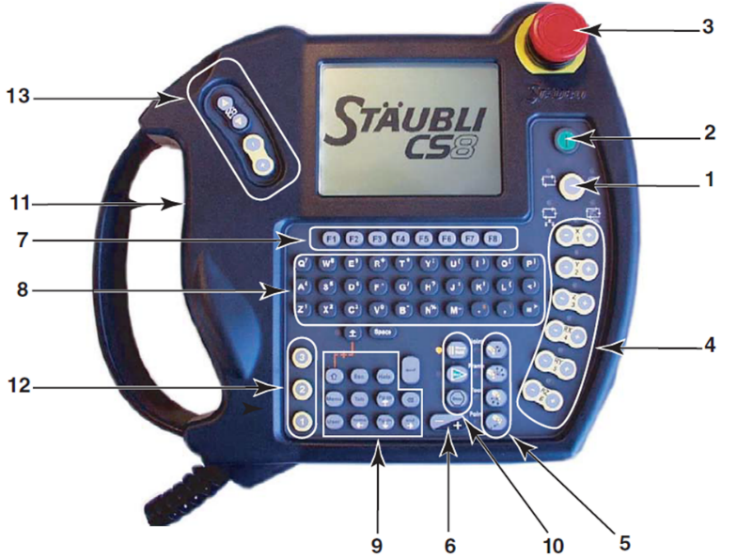
Le contact « Homme mort » à 3 positions :

1-coupe la puissance

2-autorise la mise sous puissance

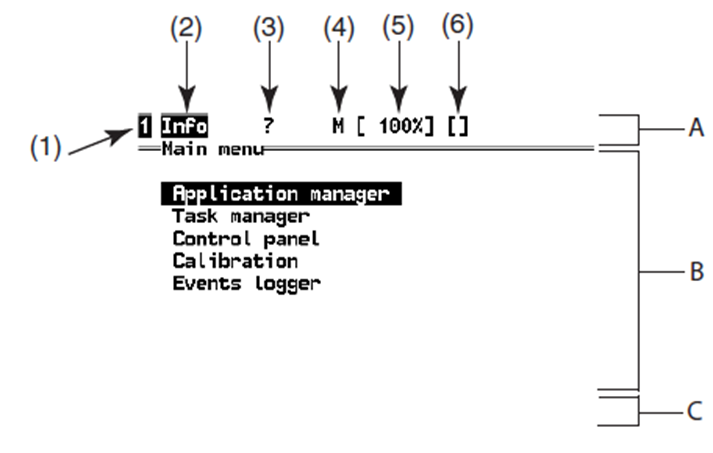
3-coupe la puissance

**Description du MCP**

Mode de marche

1. Bouton de mise sous puissance du bras
2. Arrêt d’urgence
3. Touche de mouvement
4. Touche de choix du mode de déplacement : Joint / Frame /Tool / Point
5. Touche réglage de la vitesse
6. Touches de menus contextuels
7. Touches alphanumériques
8. Touches d’interface et de navigation
9. Touches de commande des applications
10. Bouton de validation
11. Touche d’activation de sorties digitales **: *associé à la pince par exemple pour l’ouverture ou la fermeture***
12. Touche de jog

**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 :

### STAUBLI Moteurs des bras 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)

## STAUBLI : Câblage des Matériels

**/Contrôleur CS8** : **Connecteur J110** 🡺 MCP

**Connecteur J204** 🡺 ETH (@IPfixe : 10.16.7.18)

**Connecteur J113** 🡺 WMS

**Connecteur J109** 🡺 Chaine des AU géré par PILZ (PnozMULTI)

**Connecteur FO** avec 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*

**/BrasRobotTX40** : **Connecteur P1** 🡺 Alimentation Electrovanne

**Connecteur FO** 🡺 Liaison contrôleur CS8

## STAUBLI : Modes de Marche

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

1. **Arrêt d’Urgence (AU) Système** :

**⮱ Barrière Immatérielle (BI)** du système « DEBOXEUR » et « ENCAISSAGE »

Avec les Barrières immatérielles PILZ

* 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 les BI et reseter les messages d’erreurs du système

⮱ 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. **Arrêt d’Urgence (AU) Robot**

* Pour mettre le bras sous puissance, il faut **vérifier la chaîne d’arrêt d’urgence :**

Situation d ’arrêt d ’urgence :

⮱ 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. **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. **CS8 :** Tourner le sélectionneur (6) en position ON

**Indicateurs visuels :**

⮱ 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**

### 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)

1. **Vérifier que la cellule robotique est mise en énergie**
2. **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 (MANU) :**

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.

## Relier B&R à STAUBLI 🡺 .xdd : pour une liaison EPL

### Comment trouver le fichier .xdd ?

**NB** : *Pour certain fabricant, vous trouverez le fichier .xdd sur le site internet de celui-ci, vous choisissez alors le fichier xdd correspondant à notre module.*

Pour Staubli, nous trouverons le fichier **.xdd** dans la CPU STAUBLI : le CS8c

**Prérequis** : Pour récupérer le fichier XDD vous devez allumer le robot afin de le récupérer dans la CPU du robot

A picture containing text, screenshot, design

Description automatically generated

1. Lancer le logiciel SRS 2022
2. SRS : Sélectionner l’application / onglet Accueil (Home) / IO physiques

A picture containing text, design

Description automatically generated

1. SRS : (onglet Général /) Ajouter une carte I/O
2. SRS : Choisissez le protocole Powerlink / OK
3. SRS : Sélectionner « Hilscher »

A picture containing text, font, graphics, screenshot

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

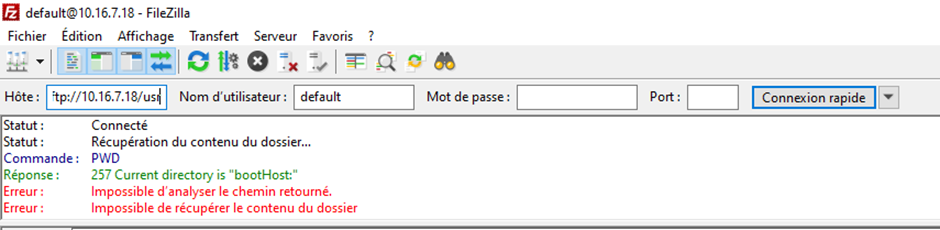
1. SRS : Ajouter un module
2. SRS : Choisissez le nombre d’entrées et de sorties digitales souhaités.

A screenshot of a computer

Description automatically generated with medium confidence

1. Maintenant nous devons récupérer le fichier xdd dans le CS8, nous ferons ça avec FileZilla.
2. Dans FileZilla : Hôte : **ftp://10.16.7.18/usr**

Identifiant : **default**

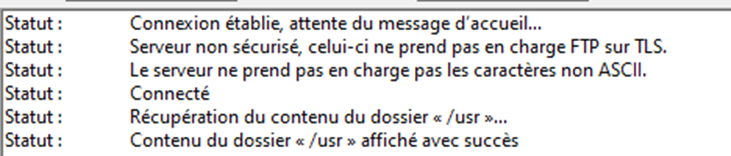
Laisser les autres champs vide

Une image contenant texte, capture d’écran, affichage, logiciel

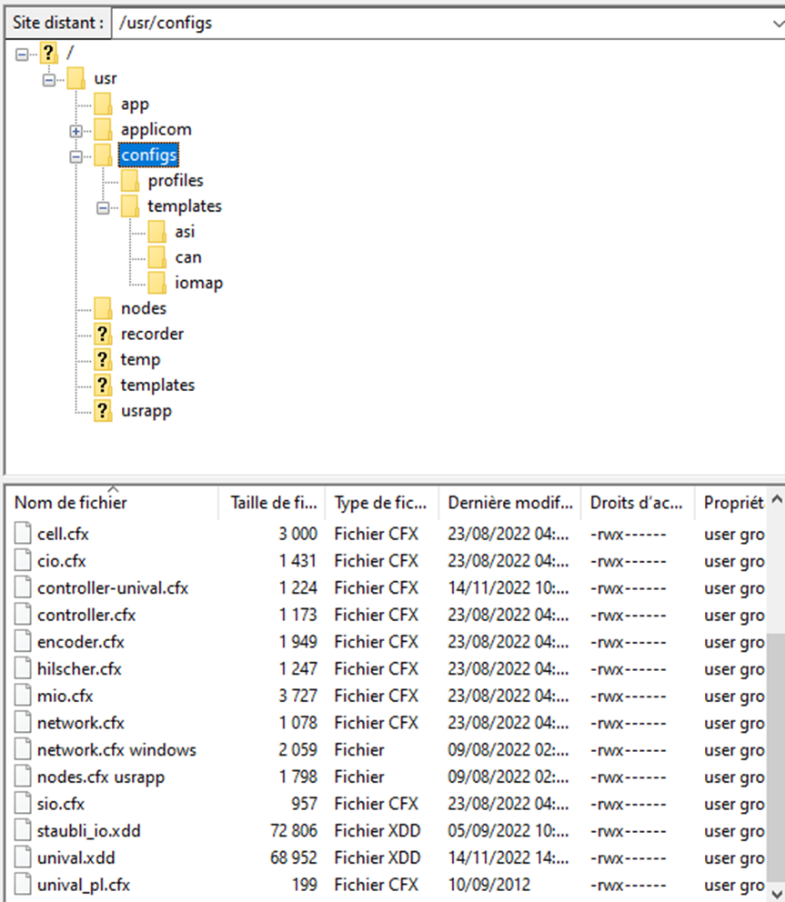
Description générée automatiquement

⮱ Connexion rapide

⮱ Une notification vous dira qu’il est déjà connecté au serveur mettre interrompre après ceci le status se mettra à jour

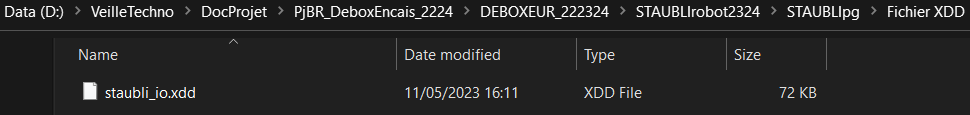


1. Dans le CS8, dans le dossier **usr/configs**, il y a le fichier **staubli\_io.xdd**



⮱ Sur site à distant copier ouvrir le fichier /usr/configs plusieurs fichiers seront affichés télécharger staubli \_io.xdd

1. Transférez-le dans le dossier de votre choix, ça y est vous l’avez !



### Importer le fichier .xdd dans la CPU B&R

1. BR / Créer un projet AS: **DeboxBrStaubli.apj**
2. BR / Hardware :

CPU B&R : PPc70 / ETH = 10.16.7.34 / **EPL** avec le nœud = **2** (à vérifier dans AS)

CPU STAUBLI : CS8c / ETH = 10.16.7.18 / Carte **EPL** avec le nœud = **3** (à vérifier sur la roue codeuse)

1. BR / Software :

AS **4.6.3.55** (pour être compatible avec le projet « Encaisseuse »)

AR **B4.62**

1. A picture containing text, screenshot, rectangle, computer

   Description automatically generatedBR / Synoptique :

A screenshot of a computer

Description automatically generated

1. BR / Physical View / **PLK** : clicG
2. BR / Sélectionner l’onglet **Tools** / **Manage 3rd Party Device** / **Import Fieldbus Device**, sélectionner le fichier **.xdd** puis ouvrir.
3. BR / Sélectionner PLK (Powerlink)
4. A screenshot of a computer

   Description automatically generatedDans la partie droite de l’écran, dans le Catalog, double cliquer sur le fichier .**xdd**
5. Résultat

NB : Vérifier que la CPU Staubli est alimentée en énergie

NB : Vérifier que le module supervised = off pour la carte « staubli-io »

A screenshot of a computer

Description automatically generated

1. Vérifier le Hardware avec le SDM :

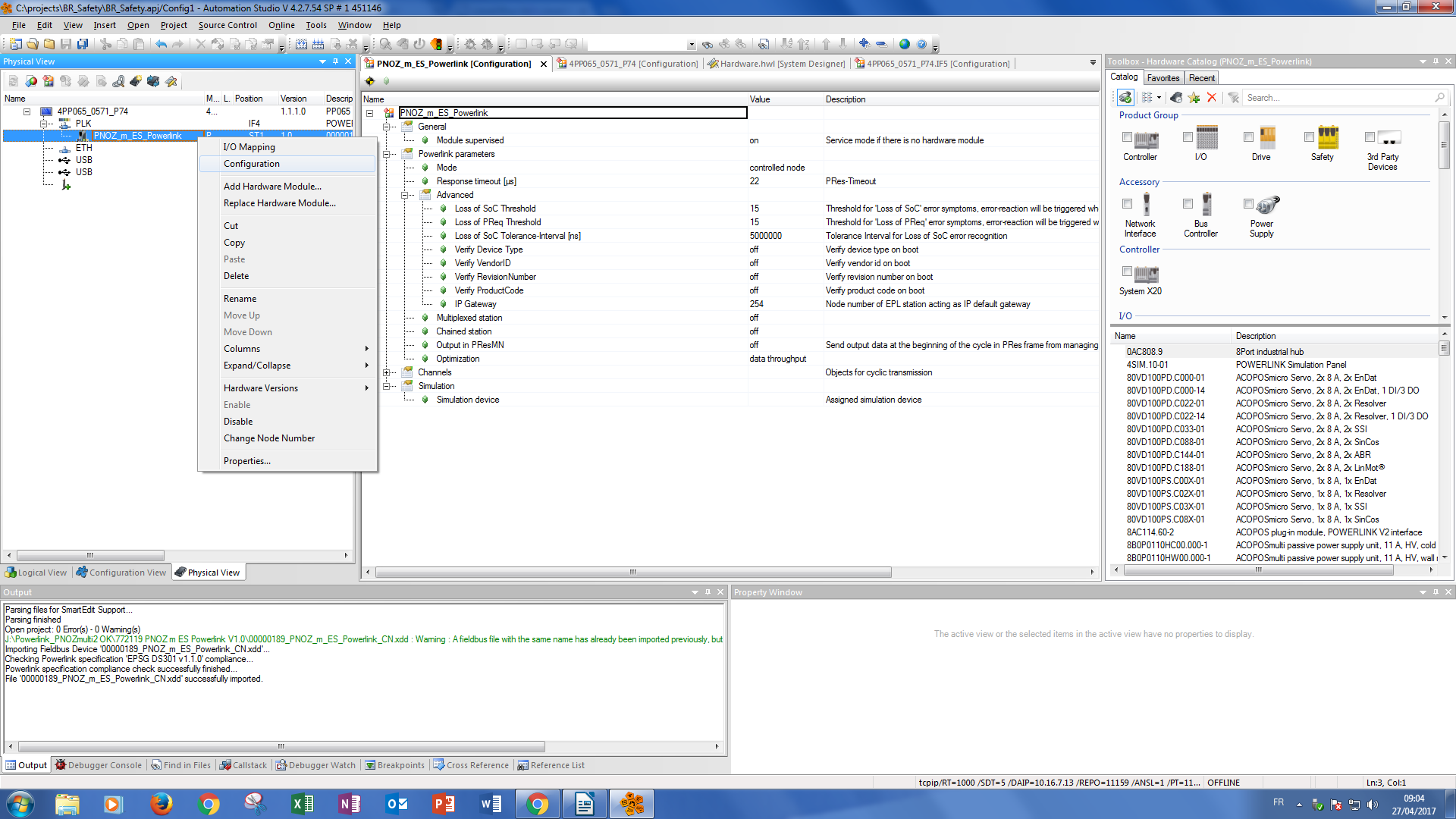
A screenshot of a computer

Description automatically generated

### Configuration du module importé par le fichier xdd : Com B&R / STAUBLI

BR / Faites un clic droit sur le module importé (ici le module Powerlink : **staubli\_io**) puis configuration

⮱ Vérifier que le mode supervisé est en **off**

BR/ Puis sélectionner Channels

2

A screenshot of a computer

Description automatically generated

3

1

4

Question 1 Read…OCTET[11] ? et 1 Write… OCTET[4] ?

BR / Vérifier la présence des modules concernés entre :

⮱ Les entrées numériques : \_1\_ReadDigitalInput1\_bit\_I6020

⮱ Les sorties numériques : \_1\_WriteDigitalOutput1\_bit\_I6220.

Ici nous voulons envoyer une information de la CPU BR au robot Staubli avec la carte Powerlink du CS8c

⮱ Choix de la carte \_1\_WriteDigitalOutput1\_bit\_I6220 avec la Cyclic transmission = **Write**.

BR / Visualiser le mapping : Clic droit (ici) sur « staubli\_io » puis **I/O Mapping**.

⮱ Nous pouvons alors créer notre communication powerlink avec l’entrée du CS8c.

A screenshot of a computer

Description automatically generated with medium confidence

1

A screenshot of a computer

Description automatically generated with low confidence

2

* **Ecriture du programme**

BR / Sélectionner **Logical View** puis placer vous sur le nom du projet : **DeboxBrStaubli**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated with medium confidenceBR / Sélectionner Program puis **ST Program All In One**

**BOOL\_Bouteille\_1** qui enverra l’état du futur bouton de l’IHM **BP\_Bouteille\_1** sur le powerlink. (Information sur 1 bit)

NB : Les variables se déclarent automatiquement.

Pour cela double cliquez sur Program puis sur CYCLIC et nous pouvons écrire la ligne de code dans PROGRAM\_cylic.

A screenshot of a computer

Description automatically generated with low confidence

NB : Il faut vérifier le type de BOOL\_Bouteille\_1 qu’il faut mettre en BOOL.

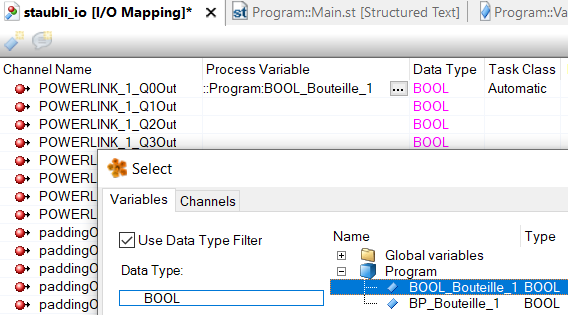
A screenshot of a computer program

Description automatically generated with medium confidence! le type demandé par la communication (/ staubli\_IO configuration / Channel : OCTET[4]).

Pour cela double cliquez sur Variables.var et changer le type INT en USINT ou BOOL.

BR / Puis on rebuild.

Maintenant nous pouvons mettre la variable BOOL\_Bouteille\_1 dans la communication powerlink. Pour cela retournons dans Physical View puis I/O Mapping du module staubli\_io. Cliquez sur le bouton pour ajouter une variable et sélectionner notre variable BOOL\_Bouteille\_1 et OK.



Voilà notre communication est prête (voir AS Visu pour la création de l’IHM).

* **Simulation de la connexion avec le monitoring d’AS B&R**

Nous pouvons simuler le bouton en se mettant en Monitoring puis en sélectionnant ForceActivated et en écrivant 0 ou 1 dans la case ForceActivated Value.

A screenshot of a computer

Description automatically generated

**En effet, le bit envoyé sur une sortie EPL de la CPU B&R est réceptionnées sur une entrée EPL de la CPU STAUBLI**

⮱ Sur le boitier Manuel **MCP**, nous pouvons voir les mvt des variables IO en temps réel

⮱ Control Panel / **+I/O** / **+Powerlink** / **+Powerlink-1-Dinputs** :

I0= **off** ou **on**

I1= **off** ou **on**

**…**

**STAUBLI / MCP :** Résultat

A screen shot of a computer

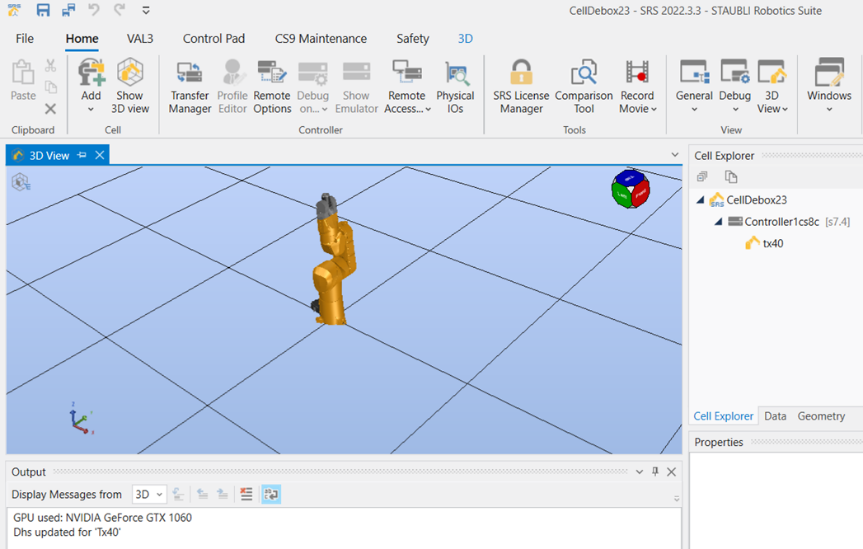
Description automatically generated with low confidence A screen shot of a computer

Description automatically generated with low confidence

# STAUBLI Programmation

## Logiciel de développement SRS 2022 :

A screenshot of a computer

Description automatically generated with low confidence

La programmation du robot se fait grâce au logiciel SRS 2022. Ce logiciel permet de coder l’algorithme du robot en langage VAL3 qui est un langage de programmation industriel. Ce logiciel permet de simuler en 3D les mouvements du robot ainsi que le pupitre de commande du robot en émulateur. C’est utile pour tester le programme avant de l’exporter sur le véritable robot.

A screenshot of a computer

Description automatically generated

**Emulateur du pupitre de commande dans SRS 2022**

## Le programme VAL3 avec SRS

Les mouvements se font par de simples lignes où il suffit de renseigner le point voulu, l’outil et la vitesse. Après chaque mouvement, il faut insérer « waitEndMove » pour lui dire d’attendre la fin du mouvement avant d’exécuter le mouvement suivant. Avant de faire fonctionner le programme sur le véritable robot, il faut apprendre les points car le repère du robot n’est pas le même que celui du programme.

## La communication POWERLINK

Pour que le programme puisse communiquer avec l’affichage, on doit ajouter la carte powerlink dans le logiciel, y ajouter le nombre d’entrées/sorties voulu. Ensuite, il faut redémarrer le CS8 et récupérer le fichier xdd avec FileZilla.

Ce fichier xdd sert à mettre en place la communication avec l’ordinateur de contrôle de l’IHM avec la méthode maître‐esclave. **L’ordinateur de l’IHM est maître et le robot est esclave**. Une fois ce fichier récupéré, on peut dans le programme, lier ses entrées/sorties de la communication powerlink avec des variables que l’on utilise dans le programme où il suffit de changer sa valeur pour que le maître le reçoive.

Robot STAUBLI

Carte I/O

*READ*

CPU /

Screen

B&R

**Maître**

Carte I/O

*WRITE*

Caméra

COGNEX

*WRITE (EIP)*

## Sauvegarde des données avec une cellule STAUBLI : SRS ↔ CS8c

**⮱ SRS / Importance du Transfert Manager :** transfert **CPU/PCsrs** ou **PCsrs/CPU** :

**Procédure de Transfert du Robot / CS8c vers le PC**

ROBOT (**MCP**)

* Arrêter l’application (Run puis Stop **avec MCP**)
* F8 : Enregistrer / F4 : Fermer / F3 : Exporter (versionnage) **avec MCP**
* **Transfert manager** sur SRS
* Recharger avec SRS : /**VAL3** / **Reload Application** (*rafraichir / refresh*)

PC (**SRS**)

**Procédure de 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 :

/Application Val3 /+Pjxxxxx/ **Rech**(F5) /Oui

ROBOT (**MCP**)

## Programmer avec SRS : Procédure exemple avec le déplacement du robot sur 6 points

### Contexte

**Ecrire un programme VAL3 pour réaliser une trajectoire avec des segments en lignes droites et courbes.**

1. Créer et renseigner l’outil « tPince »
2. Créer et apprendre le joint « jDepart »
3. Créer et apprendre les points « pA » à « pC », « pPrise » et « pPose »
4. Ecrire le programme de mouvement réalisant la trajectoire

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

**Exécuter le programme via une commande extérieure : BP avec liaison Ethernet PowerLink**

Cahier des charges :

Nom de la cellule : **CellDebox2324** Points : **jDepart** Variables : **tPince**

Nom du contôleur : **Controller2324 pPrise mLent**

Version : **S7.4 pPose dDiepl** *(à modif : dIepl)*

Nom de l’application : **PjDebox pA à pC**

Nom du programme : **PgDebox6pts**

### Procédure Exemple avec le programme : PgDebox6pts

 Apprentissage des points (! **Pour apprendre un point, IL FAUT AVOIR DEFINI UN OUTIL !)**

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. **SRS /** *Paramétrer la Cellule avec son* ***nom******de la cellule =******CellDebox2324*** *avec le CS8c et le TX40*

*(Voir Guide « TutoStaubli\_UserSRS »)*

**⮱ Transfer Manager du PCdev vers le contrôleur CS8c** 🡺 A red triangle with a white exclamation mark

Description automatically generated Suivre la procédure ci-dessus

1. **MCP** **/** *Créer le* ***Nom de l’Application= PjDebox*** *et* ***l’outil=tPince*** *fixé au Flange*

Menu Principale :

**Gestionnaire d’application**

**/ 🡪** *(pour valider le chemin)*

**/ Application VAL3**

**/** F8 : **Ok** / F8 : **Nou** ou F7 : **Ouv**

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

(NB : Majuscule : appuyer simultanément sur 3 touces du MCP lettre voulu comprise)

**/ +Disque / 🡺**

**/ +Pjdebox** (surligné une fois sélectionné dans la liste)

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

**/ 🡪**

**/ +Pjdebox (date)** (/F5 : **Rech** si issue d’un transfert SRS / F8 : **oui**)

**/ 🡪**

**/ +Variables globales**

**/ 🡪**

Icon

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

**/ +Flange** : **tPince** /F8 :**Ok**

**Tableau**

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

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

**Z=↵ 85** **mm** **RZ=**0

* **Z=85 🡺 lors des tests, le robot exécutera les mouvements appris avec un recul de 85mm**

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

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

Icon

Description automatically generated ***SRS : /Physical IOs/ UserIO /Digital Outputs /valve1 (PjDebox : tPince[0])***

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

**/** F8 : **Ok**

1. **MCP / *Apprentissage des points***

⮱ Icon

Description automatically generated Plusieurs méthodes pour enregistrer un point :

**/ Nou** 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

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

Icon

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

Icon

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

**/ 🡪**

Nom = j**Depart**

**/** F8 : **Ok / tableau /** F8 : **Ok**

**MCP / mode Joint** :



X/J**1** = Tronc **+** poing mainD et pouce vers le haut

Y/J**2** = Epaule : **+** vers le bas

Z/J**3**= Coude : **+** vers le bas

RX/J**4** = Poignet : **+** poing mainD et pouce vers l’extérieur

RYJ**5** = Main

RZ/J**6** = Outil

*Effectuer un mvt sur le point choisi* ***en mode joint****:*

**/ Move :** (F1) / selection de l’outil : tPince (ici)

**/ 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) : pour apprendre des points

Icon

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

**/ 🡪**

Nom = p**A à pC /** F8 : **Ok /** F8 : **Ok**

p**Prise /** F8 : **Ok /** F8 : **Ok**

p**Pose** **/** F8 : **Ok /** F8 : **Ok**

*Effectuer un mvt* ***sur le point choisi******en mode frame*** *(règle des 3 doigts de la main Droite)*

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

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

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

MCP : Résultats des points d’apprentissages

A screen shot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generatedjDepart pPrise : pPose

A screen shot of a computer screen

Description automatically generated

A close-up of a computer screen

Description automatically generatedA close-up of a computer screen

Description automatically generatedpA pB pC

A computer screen with text and numbers

Description automatically generated

1. **MCP / *Créer variable mLent : Vitesse*** *= 30% / blend=****off****)*

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

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

**/ +mdesc**

**/** F7 : **Nou**: Nlle variable **/ OK** ou **/ 🡪** si déjà enregistré

**/ mlent /** F8: **Ok**

Vitesse =  **↵** **30** %

Blend = **↵** **Off** **/** F8:**OK**

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

⮱ **Transfer Manager** du contrôleur CS8c vers le PCdev

A screenshot of a computer

Description automatically generated

1. **SRS / *Déclarer*** *ou vérifier la présence des* ***variables***

**dDiepl** : Data entrée Powerlink

**SRS** / *Création de la variable Powerlink (pour déclencher le programme VAL3)*

/ Fenêtre **Data** / CellDebox2324

/ PjDebox : ClicD **Add / New Data** 🡺 popup : **dio**

/ name : **dDiepl** / **OK**

**mLent** : Vitesse du mouvement à 30%

⮱ *a été créée avec le MCP et nous vérifions sa présence dans SRS*

1. **SRS / Mapper les variables aux E/S**

/ Fenêtre **Data** / CellDebox2324

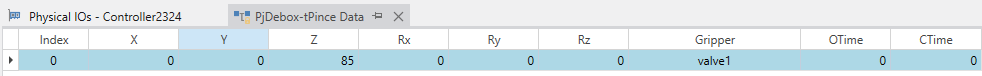
/ PjDebox  / dio / dDiepl / **dDiepl[0]**

**⮱** glisser/déposer dans la fenêtre Physical IOs / Powerlink /POWERLINK-1 / Digital Inputs / %I0

A screenshot of a computer

Description automatically generated

Même procédure pour relier la variable **tPince[0]** avec la **valve1** si cette dernière n’a pas été mappée automatiquement (à vérifier)



A screenshot of a computer

Description automatically generated

1. **SRS /** Editer le programme : **PgDebox6pts**

* *Move****j :*** *mvt courbe*
* *Mov****l :*** *mvt ligne droite*
* *Choisir 6 points plutôt que 5 pour obliger le bras à passer la barrière du convoyeur*

**begin**

wait(dDiepl==true) *//attente de l’appui touche sur PanelBR*

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

move**j**(pA, tPince,mlent)

A screenshot of a computer program

Description automatically generatedmove**l**(pPrise, tPince,mlent)

**waitEndMove()** Icon

Description automatically generated*car le robot doit aller*

move**l**(pA, tPince,mlent) ⮱ *jusqu’au point* ***pPrise***

move**j**(pB, tPince,mlent) Icon

Description automatically generated*Assurer le passage de la barrière*

move**j**(pC, tPince,mlent)

move**l**(pPose, tPince,mlent)

**waitEndMove()** Icon

Description automatically generated*car le robot doit aller:* ***point de pose***

move**l**(ppC, tPince,mlent) ⮱ *jusqu’au point* ***pPose***

move**j**(ppB, tPince,mlent)

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

**waitEndMove()** Icon

Description automatically generated *car le robot doit aller jusqu’au point jDepart :* ***point de chaîne***

**end**

Icon

Description automatically generated**waitEndMove()** : *car la scrutation est trop rapide ⇨ le pg ne se fait pas*

A screenshot of a computer

Description automatically generatedIcon

Description automatically generated Mettre PgDebox6pts dans **Start :**

**begin**

call **PgDebox6pts()** Icon

Description automatically generated au parenthèses

**end**

*NB : Ajouter les commandes de la pince (Voir guide Formation Staubli p32)*

**open(tPince)**

**close(tPince)** à placer entre les movel

*NB : modifier le pg avec les approches :* ***appro****() (voir guide MCP / Voir guide Formation Staubli p80 (ex) et p7)*

1. **SRS /** tester le programme : **PgDebox6pts en mode simulation**

A screenshot of a computer

Description automatically generated/ **Control Pad** : Fenêtre Control Pad :

⮱ Modifier le “Step time” = **80**ms (pour SRS2022)

NB : l’émulateur avec SRS doit être lancé (être positionné sur le contrôleur dans l’arborescence)

/ ClicG sur **Start** 🡺 popup : Emulateur simulé

/ ClicG sur le Bouton de mise en énergie de l’émulateur

Emulateur simulé / **Application Val3** / Ok

Emulateur simulé / ⭘ **RUN** : ClicG/ **PjDebox** / F8 : **Ok**

Emulateur simulé / 🞊 **RUN** : ClicG

Emulateur simulé / **MOVE** (Move/hold)

⮱ Clic continu sur MOVE pour voir l’animation du bras en 3D avec le mode manuel

* **Si message d’erreur avec le robot simulé : voir la calibration initiale de celui-ci**

**⮱** Emulateur simulé /ClicG sur **Cal.** (F5) / ClicG sur **Oui** (F8)

* En simulation, peut être mettre en commentaire la ligne **//wait(dDiepl==true)** pour faire bouger le robot en simulation.

⮱ **Transfer Manager** du PCdev vers le contrôleur CS8c

1. **MCP / 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

**+Pj**Debox (surligné)

**Ok** (F8)

⮱ Indicateur Visuel MCP :

Graphical user interface, application, chat or text message

Description automatically generated Touche **RUN** : led orange Fixe

Touche **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

A picture containing electric blue

Description automatically generated✓ 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

1. **MCP / Exécuter le programme en mode automatique :**

**Procédure identique pour le mode automatique**

/sélectionner sur le WMS le mode automatique

/Appui touche sur Run 🡺 sélectionner l’application

/**Appui simple** sur la touche **Move/Hold** pour lancer les mouvements du bras robotique.

1. **°MCP / Procédure réapprentissage des points :**
2. Outil
3. Toul / frame
4. Ici
5. Test
6. **modif**

NB : Voir pour lisser les points et le mode automatique

## Programmation Déboxeur

### Problématique :

La CPU B&R envoie au robot Staubli :

* Le start et le stop du ou des mouvements (gestion des cadences)
* Les coordonnées X, Y et Z des pièces à prendre.
* …

Le CS8c exécute les mouvements avec :

* Les coordonnées reçues
* …

### Les variables utilisées dans les programmes AS et SRS

### Créer les touches de start et stop dans AS

**AS / Nom du bouton = BP\_start\_1/ clé vkReset / ToggleDatapoint (fonctionnement « interrupteur »)/ variable BR = BP\_Bouteille\_1** (Voir tuto BR Visu VC4 (/ Support / IHM))

## Maintenance :

* Vérifier les numéros des nœuds : **OK**
* Le câble entre le power panel et le CS8c est bien un câble PLK : **OK**

Même en s'affranchissant de la partie applicative l'envoi de données ne fonctionnent pas !!!

Voici le screen du mapping IO du port PLK en mode monitoring :

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidenceJ’ai bien l’impression que coté B&R le principe est fonctionnel car le module OK est à TRUE sur le staubli.

**Est-ce que coté staubli il y a du diagnostic ?**

⮱ Sur le boitier Manuel MCP, nous pouvons voir les mvt des variables IO en temps réel

⮱ Control Panel / **+I/O** / **+Powerlink** / **+Powerlink-1-Dinputs** :

I0= **off** ou **on**

I1= **off** ou **on**

**…**

**STAUBLI / MCP :**

A screen shot of a computer

Description automatically generated with low confidence A screen shot of a computer

Description automatically generated with low confidence

# Annexes

## Structure Réseau « DEBOXEUR »

A vérifier !!!

Robot STAUBLI

Carte I/O

*READ*

CPU /

Screen

B&R

Maître

Carte I/O

*WRITE*

Caméra

COGNEX

*WRITE (EIP)*

## SRS 2022

* Touche F6 = verifier le code Val3

## STAUBLI / Carte POWERLINK

⮱ Voir document constructeur : StaubliCartePowerlink.pdf

### Indicateurs visuels :

A picture containing text, screenshot, number, parallel

Description automatically generated

### A close up of a device Description automatically generatedNœud Carte STAUBLI / POWERLINK

A circular object with numbers

Description automatically generated

## Programmation exemple avec le projet « Deboxeur17 »

A screenshot of a computer

Description automatically generatedArborescence exemple sous AS :

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generatedProgrammation AS pour bouger le robot STAUBLI :

A close-up of a white background

Description automatically generated

Arborescence exemple sous SRS :

A screenshot of a computer

Description automatically generated

**AS** : Appui sur BP « Bouteille 1 »

A screenshot of a computer

Description automatically generated with low confidence

* dPowerlink**Sortie**[0] **:**

**⮱ c’est la sortie Powerlink de la CPU BR et SRS**

= consigne envoyée au robot par les CPUs.

Car CPU = Maître et Robot = Esclave

A screenshot of a computer

Description automatically generated**A screenshot of a computer

Description automatically generated with medium confidenceSRS**:

A screenshot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated

**// Je fais les mouvements pour la bouteille 1**