

FORMACIÓN VAL3

VALTRACK

S7.3.0

Stäubli Española



ÍNDICE

- ***Planos de referencia locales.***
- ***Librerías.***
- ***Hardware.***
- ***Configuraciones.***
- ***Wizard.***

FORMACIÓN VAL3

PLANOS DE REFERENCIA LOCALES

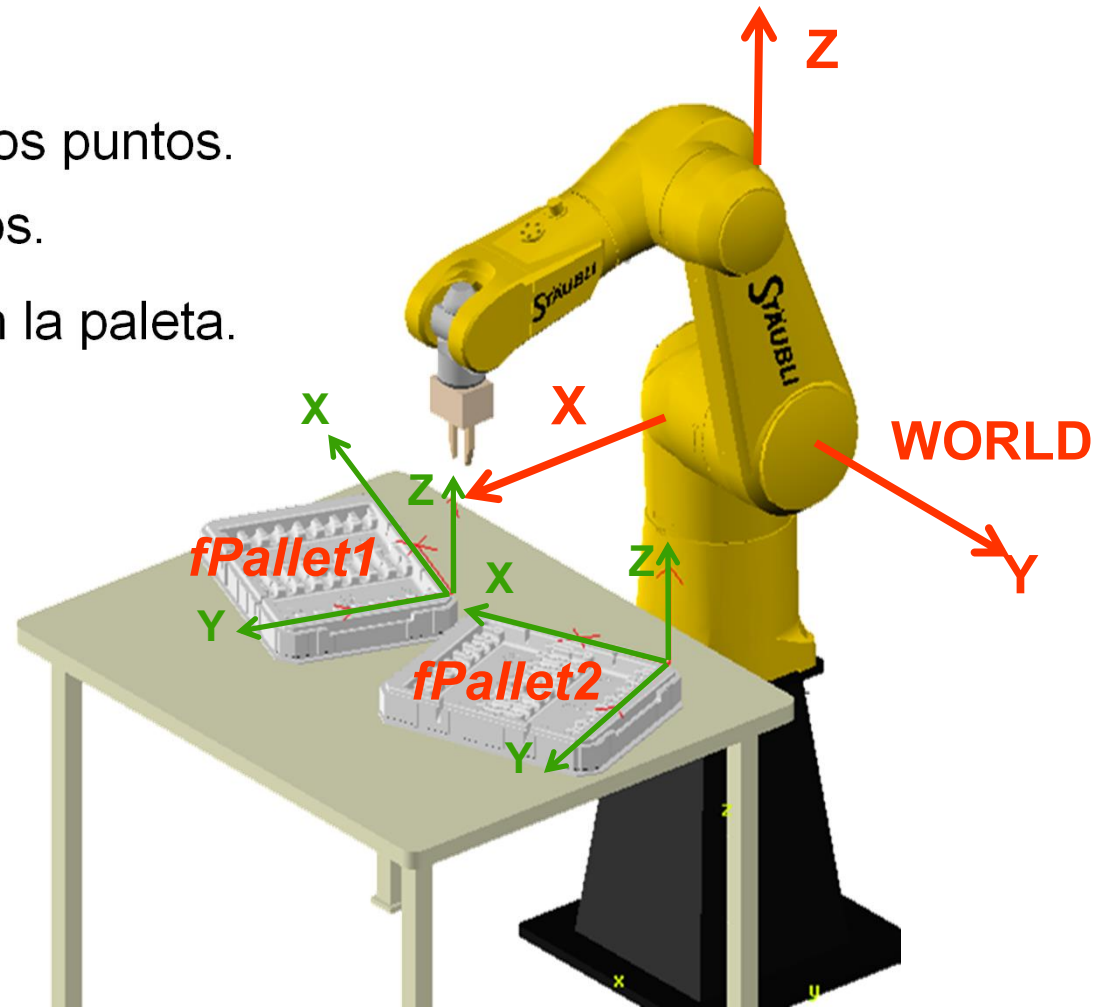


UTILIZACIÓN DE UN PLANO

Sistema de referencia local:

- Para facilitar aprendizaje de los puntos.
- Utilizados para duplicar puntos.
- Desplazamiento de puntos en la paleta.

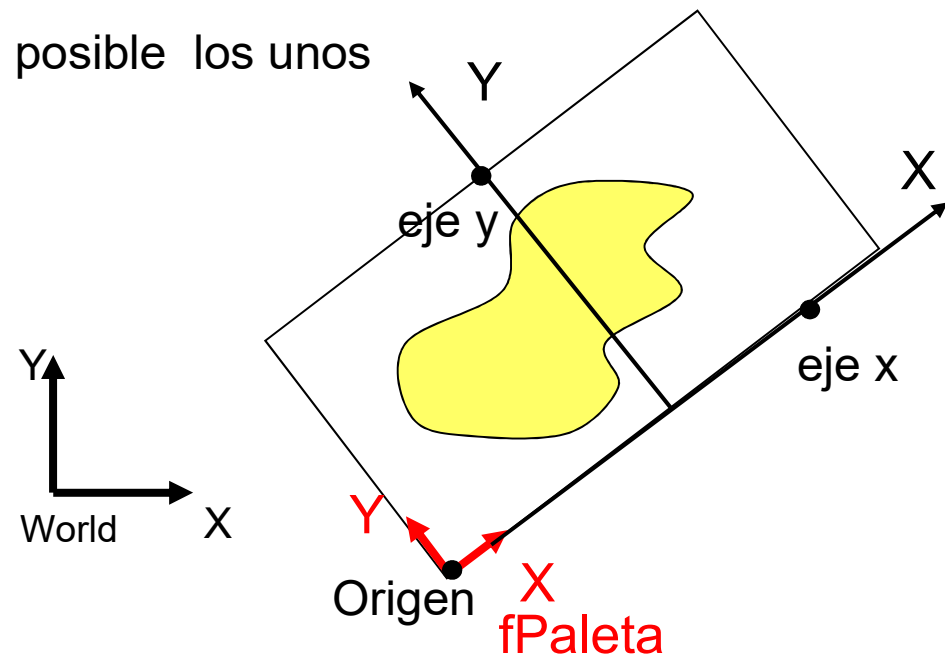
```
100%
Gestor de aplicaciones
-Variables globales
+flange
-world
-frame fPaleta
  pOrigenTomar
  pOrigenPoner
+joint
+mdesc
bool
+num
string
aio
dio
Defi Edi. Ren. Ins. Bor. Nue. Gua.
```



APRENDIZAJE DE UN PLANO (FRAME)

Definido con 3 puntos que deben aprenderse:

- Usar una herramienta precisa: Punta.
- Definir esta herramienta como la herramienta actual.
- Aprender los puntos lo más separado posible los unos de los otros (Mayor precisión).

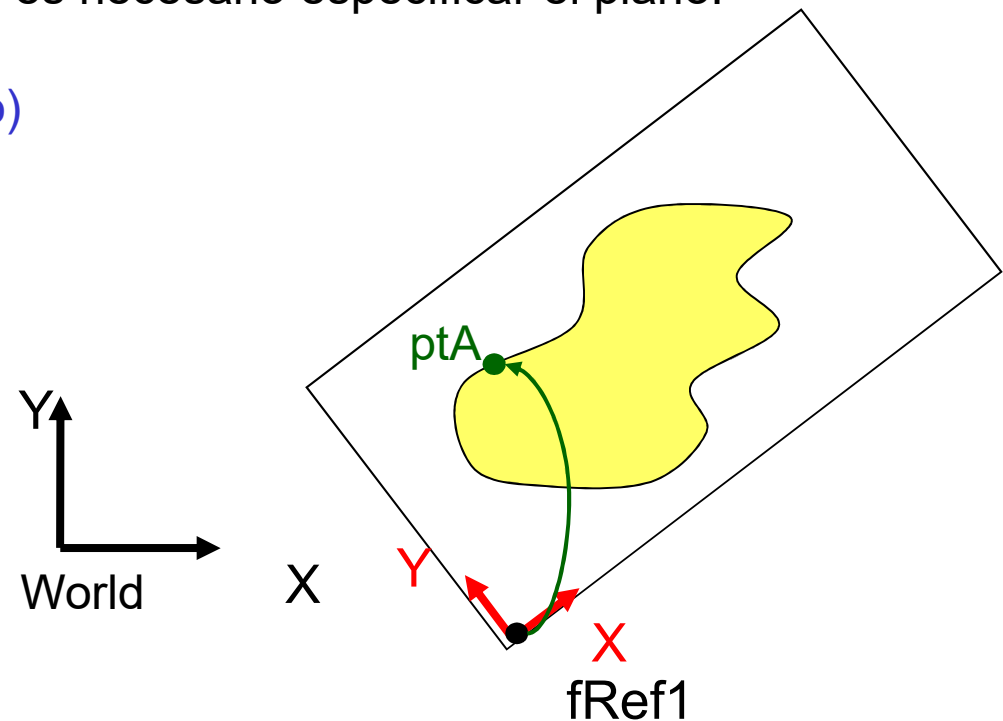
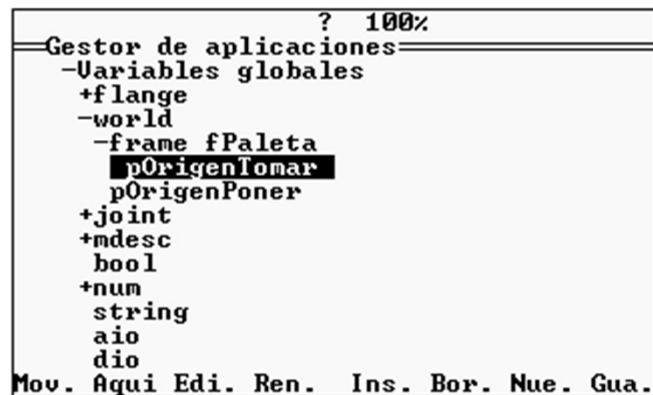


PUNTOS EN UN PLANO (FRAME)

Crear los puntos de la aplicación bajo la rama del plano:

- Durante la enseñanza del punto, las coordenadas son expresadas en referencia al plano.
- Para la instrucción del movimiento, no es necesario especificar el plano:

`movej(ptA ,tPinza, mRapido)`



FORMACIÓN VAL3



LIBRERIAS



OBJETIVO

Utilización de programas/datos en múltiples aplicaciones.

Ejemplo:

- Librerías de programas para reutilizar su código fuente.
- Librerías PIEZAS para utilizar una única aplicación con múltiples referencias de piezas.
=> 1 aplicación única + 1 librería de puntos para cada referencia de piezas.

Una librería es una aplicación normal, creada como una aplicación simple: desde el control CS8 o emulador CS8

PERO se deben definir los datos/programas que serán exportados y utilizados por otras aplicaciones.

EXPORTACIÓN DE DATOS DE LA LIBRERÍA

```
10%
==Application manager==
-Global data
+fl New data
-wo Name : pPuntoTomar
+f
+jo Type :point
+md
+bo Elements nb. : 1
+nu
st Attribut :Private
ai
+di
sio
config
Esc Ok
```

Ejemplo : El punto **pPuntoTomar** es privado y solo puede ser usado internamente a la aplicación.

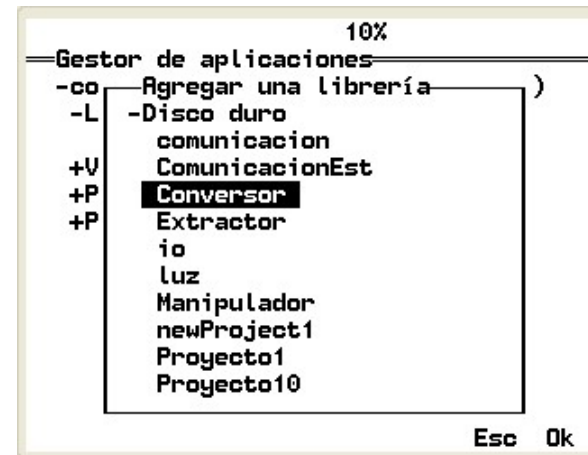
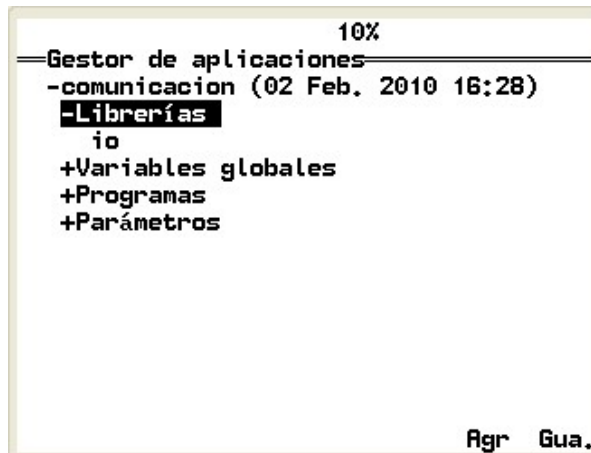
Cambia el atributo de Privado a **Público**

```
10%
==Application manager==
-Global data
+fl New data
-wo Name : pPuntoDejar_
+f
+jo Type :point
+md
+bo Elements nb. : 1
+nu
st Attribut :Public
ai
+di
sio
config
Esc Ok
```

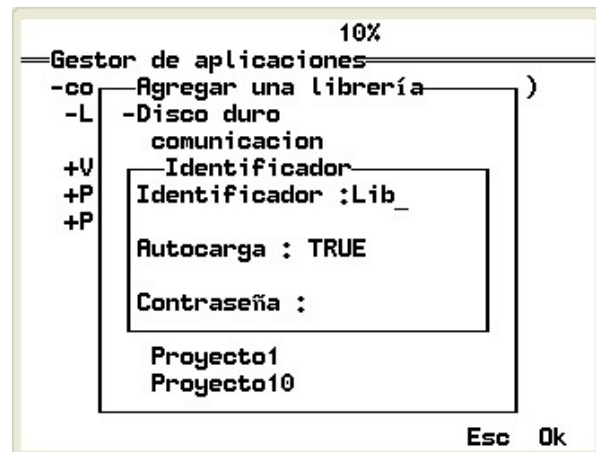
Ejemplo : El punto **pPuntoDejar** es público y por lo tanto accesible por otras aplicación.

DECLARACIÓN DE LIBRERÍAS

Las librerías utilizadas en una aplicación deben ser declaradas



Selección de la librería



El identificador se utiliza en la aplicación que importa la librería

Ejemplo : La librería Conversor es utilizada en la aplicación “comunicacion” con el identificador “lib”

UTILIZACIÓN DE DATOS

Utilización del punto pPuntoDejar definido como público en la librería `comunicacion`:

```
movej(lib:pPuntoDejar, tPinza, mRapido)
```

- Asignación de la variable numérica nX definida como pública en la librería `comunicacion`:

```
lib:nX=10
```

- Llamada al programa INIT definido como público en la librería `comunicacion`:

```
call lib:init()
```

Durante la edición del programa, el sistema garantiza la coherencia de los datos y la sintaxis de los mismos.

CARGAR LIBRERÍAS

Ejemplo: Una aplicación con múltiples librerías de punto. ptA está definido como público en ref1 y en ref2.

```
nErr=lib:libload("ref2")      carga ref2 con el identificador Lib
//Resultado de la carga en nErr, nErr=0 si se carga bien
movej(lib:ptA,tPinza,mRapido)  utiliza ptA de ref2
nErr=lib:libload("ref1")      carga ref1 con el identificador Lib
//Resultado de la carga en nErr, nErr=0 si se carga bien
movej(lib:ptA,tPinza,mRapido)  utiliza ptA de ref1
```



1 sólo identificador de librería para trabajar con varias librerías con los mismos datos exportados pero con valores diferentes

SALVAR LIBRERÍAS

libSave (): Guarda las variables y programas públicos asignados al identificador de librería.

libSave () → Guardar

libSave ("****") → Guardar como

```
nErr=lib:libSave()  
nErr=lib:libload("ref2")  
movej(lib:pA,flange,nom_speed)  
lib:nX=10  
nErr=lib:libSave()  
nErr=lib:libload("ref1")  
put(lib:nX)
```

Guarda la librería en memoria con el nombre lib
Carga la librería ref1 con el identificador lib
Usa el Pa de Ref2
Modifica NX de ref2
Guarda nX en el disco
Carga la librería ref2 con el identificador lib
Muestra el valor de nX de la librería ref1

```
nErr=lib:libSave("ref3")  
nErr=lib:libload("ref2")  
nErr=lib:libSave("ref3")
```

Guarda la librería en memoria con un nuevo nombre ref3
Copia « ref2 » to « ref3 »

ERRORES DE GESTIÓN DE LIBRERÍAS

Todas las instrucciones de librerías devuelven un valor de error. Si el valor es 0 no hay errores.

`nErr= lib:libLoad("data")`

11 : Error de carga = interfaz publica no corresponde con la librería.

12 : No se ha podido cargar librería.

`nErr = lib:libLoad("data") lib:libSave("data") libDelete("data")`

Error en la cadena asignada:

20 : Unidad no valida. (Unidad)

21 : Camino invalido. (Directorio)

22 : Nombre incorrecto.

`nErr = lib:libLoad("data") lib:libSave() lib:libSave("data")`

30 : Error de escritura / lectura (floppy, ftp,)

`nErr = lib:libSave("data")`

31 : Error de escritura = la librería ya existe (no hay posibilidad de sobreescritura → primero borrar la librería vieja)

FORMACIÓN VAL3



HARDWARE

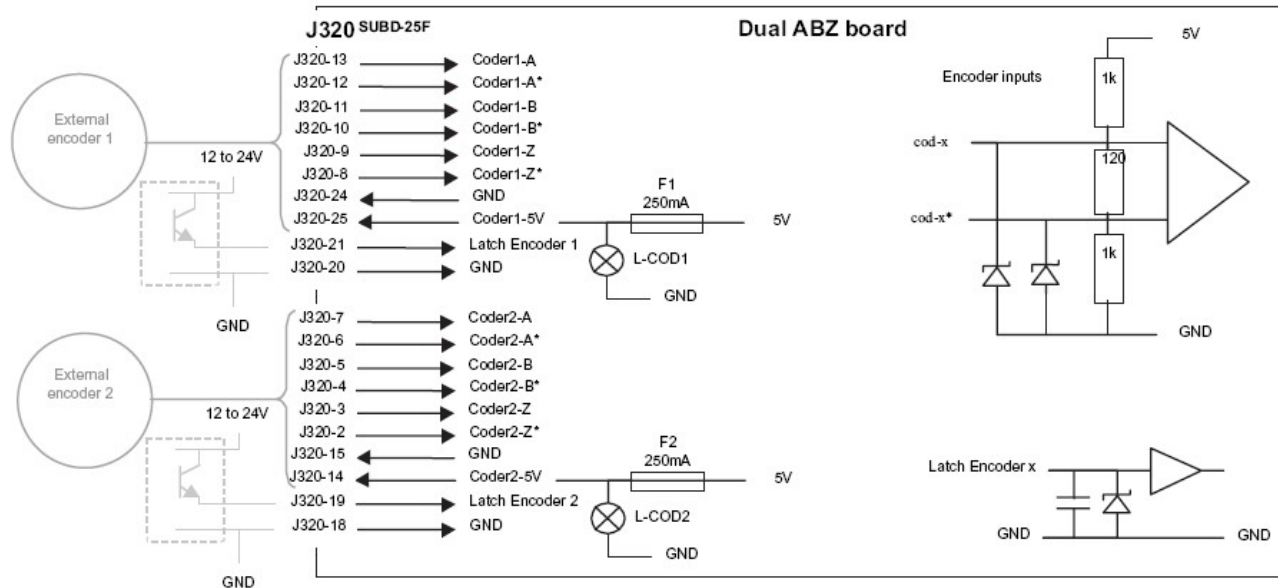


CS8C

Controlador CS8C con Starc2 y hasta 3 tarjetas lectoras de encoder (Duales ABZ)



ENCODER



$$A = A$$

$$A^* = \overline{A}$$

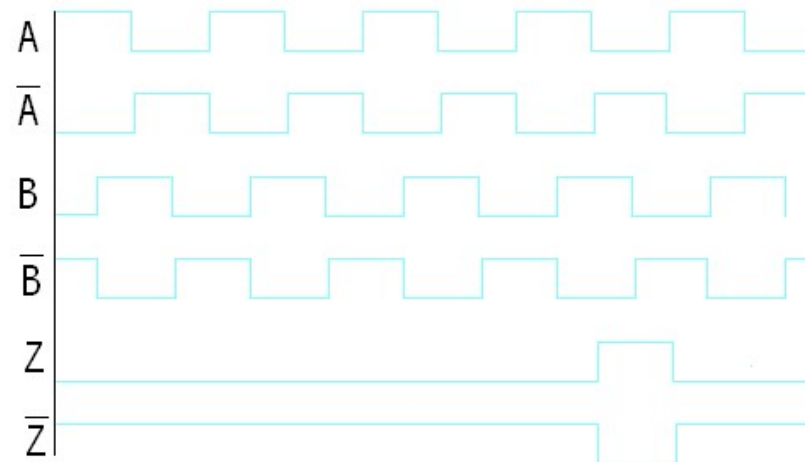
/A es A desplazada 180°.

B es A con un desfase de 90°.

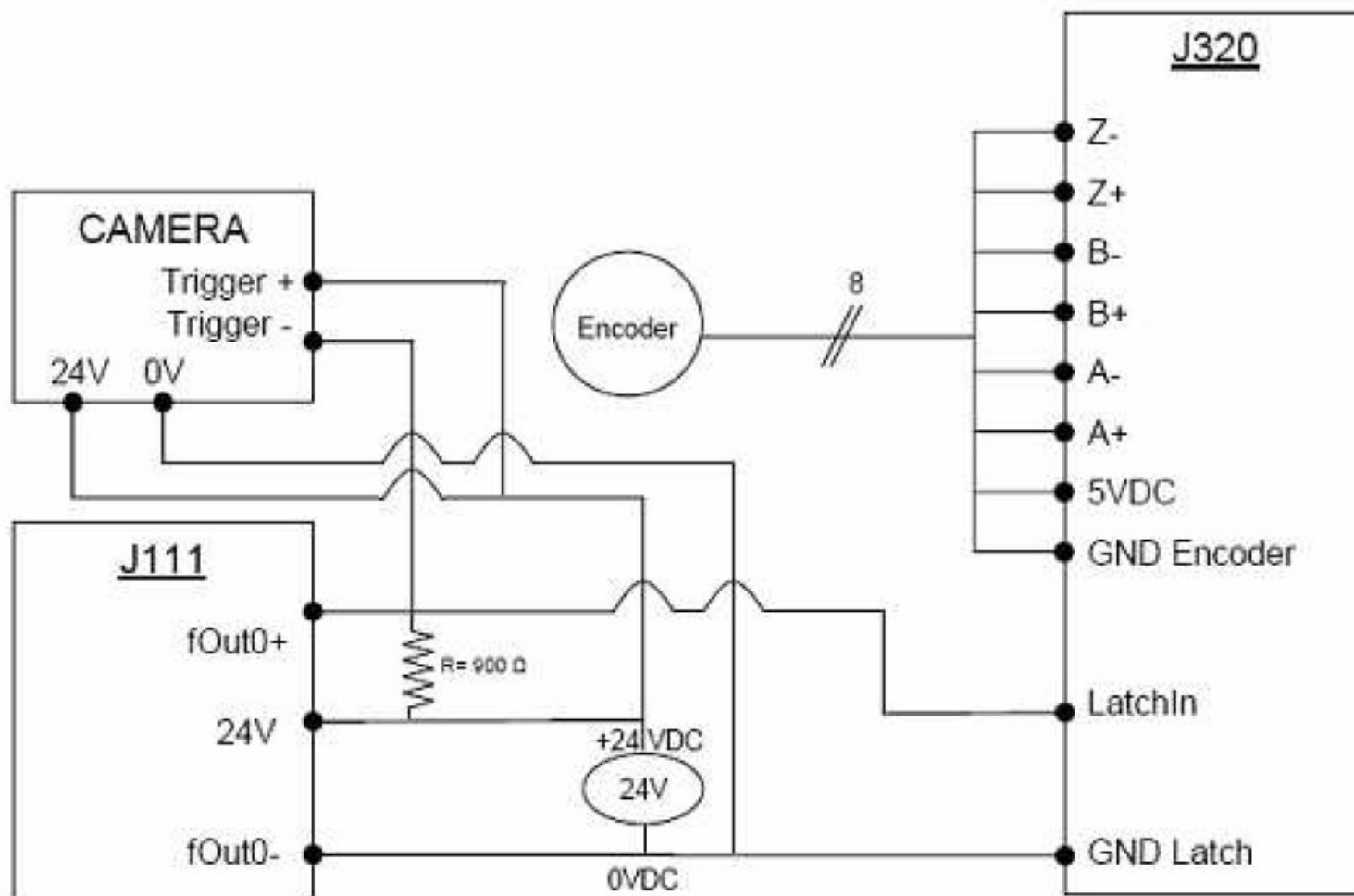
/B es B desplazada 180°.

/Z es Z desplazada 180°.

Cada vuelta de encoder se produce un pulso de la señal Z.



CABLEADO CS8C / CAMARA / ENCODER



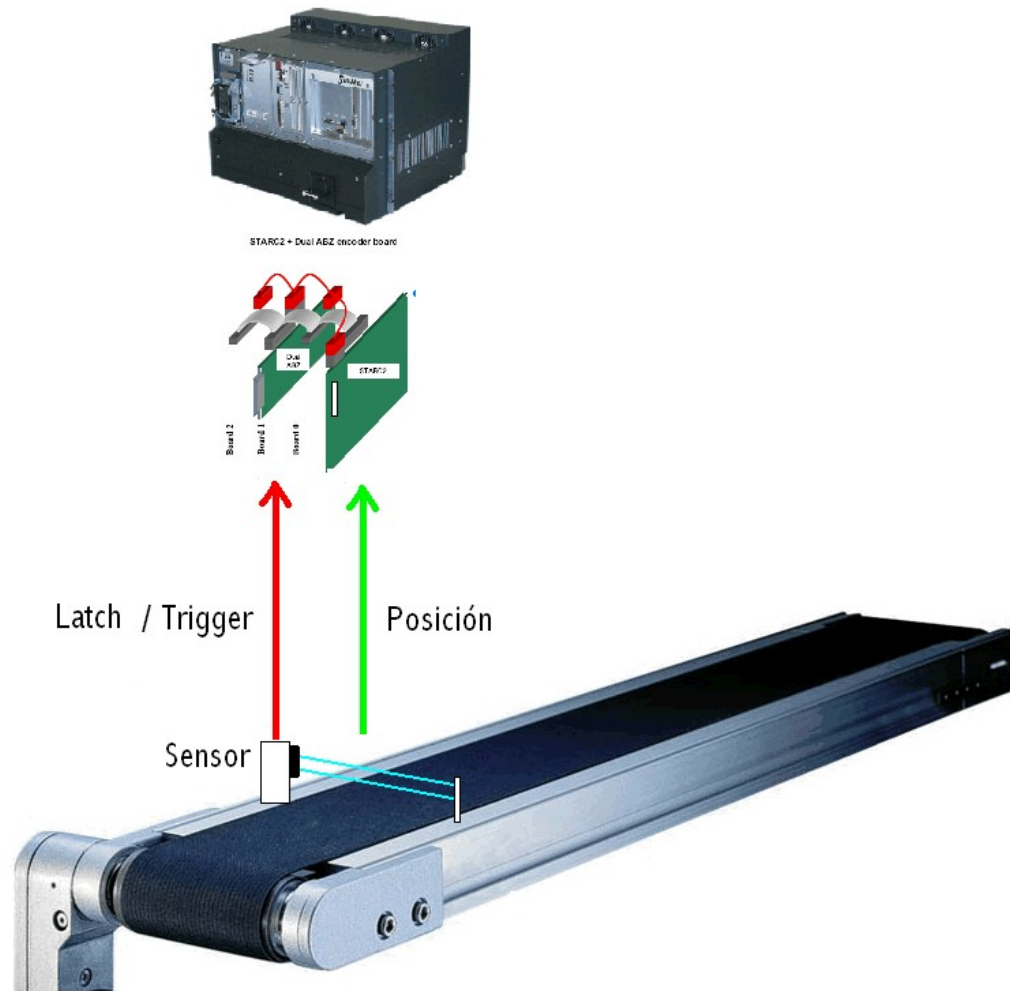
FORMACIÓN VAL3



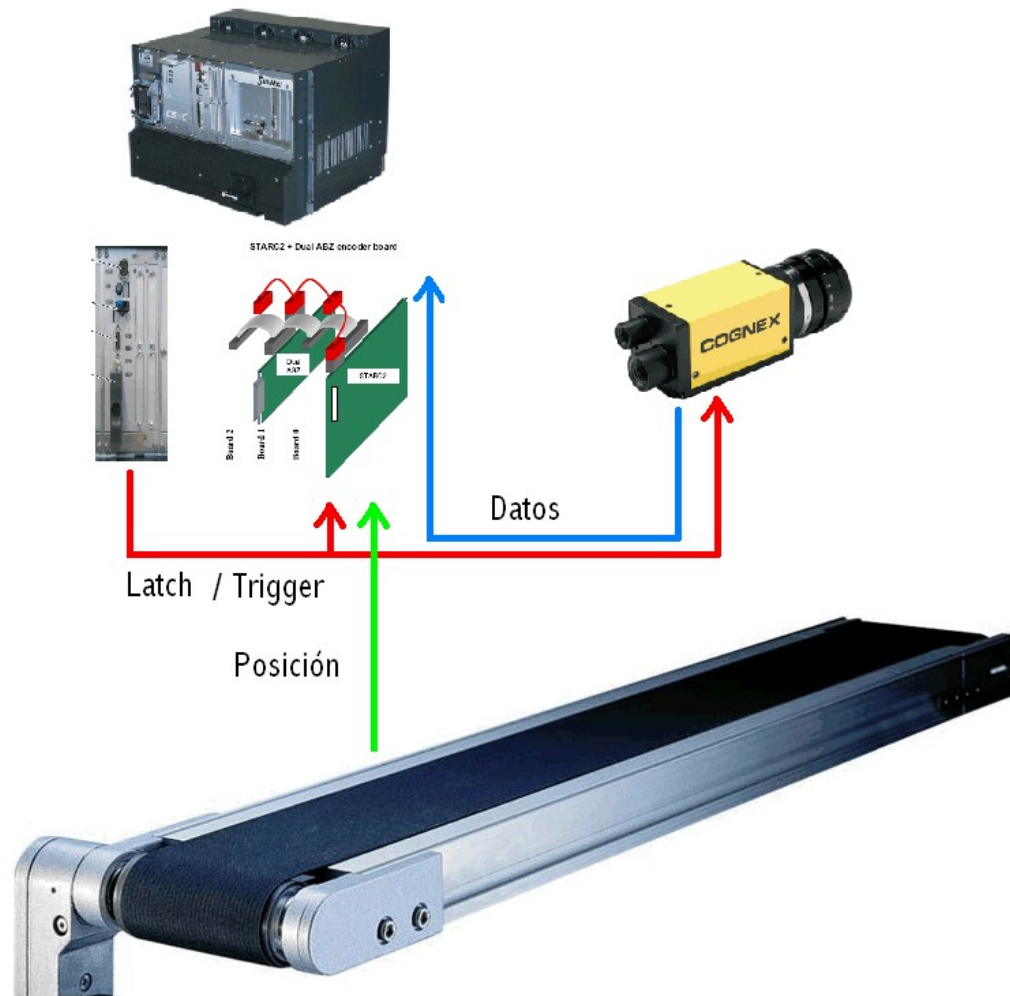
CONFIGURACIONES



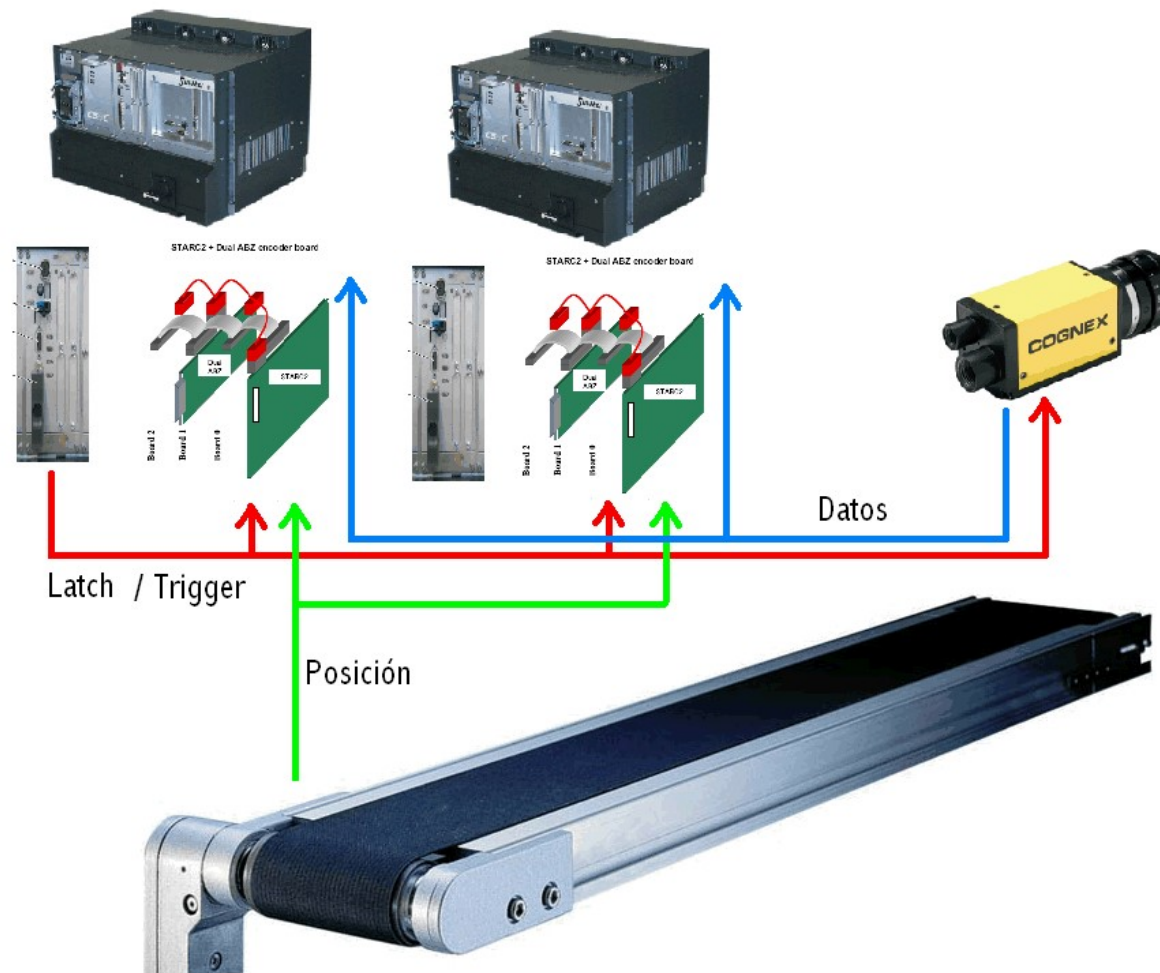
CONFIGURACIÓN 1 CS8C Y SENSOR



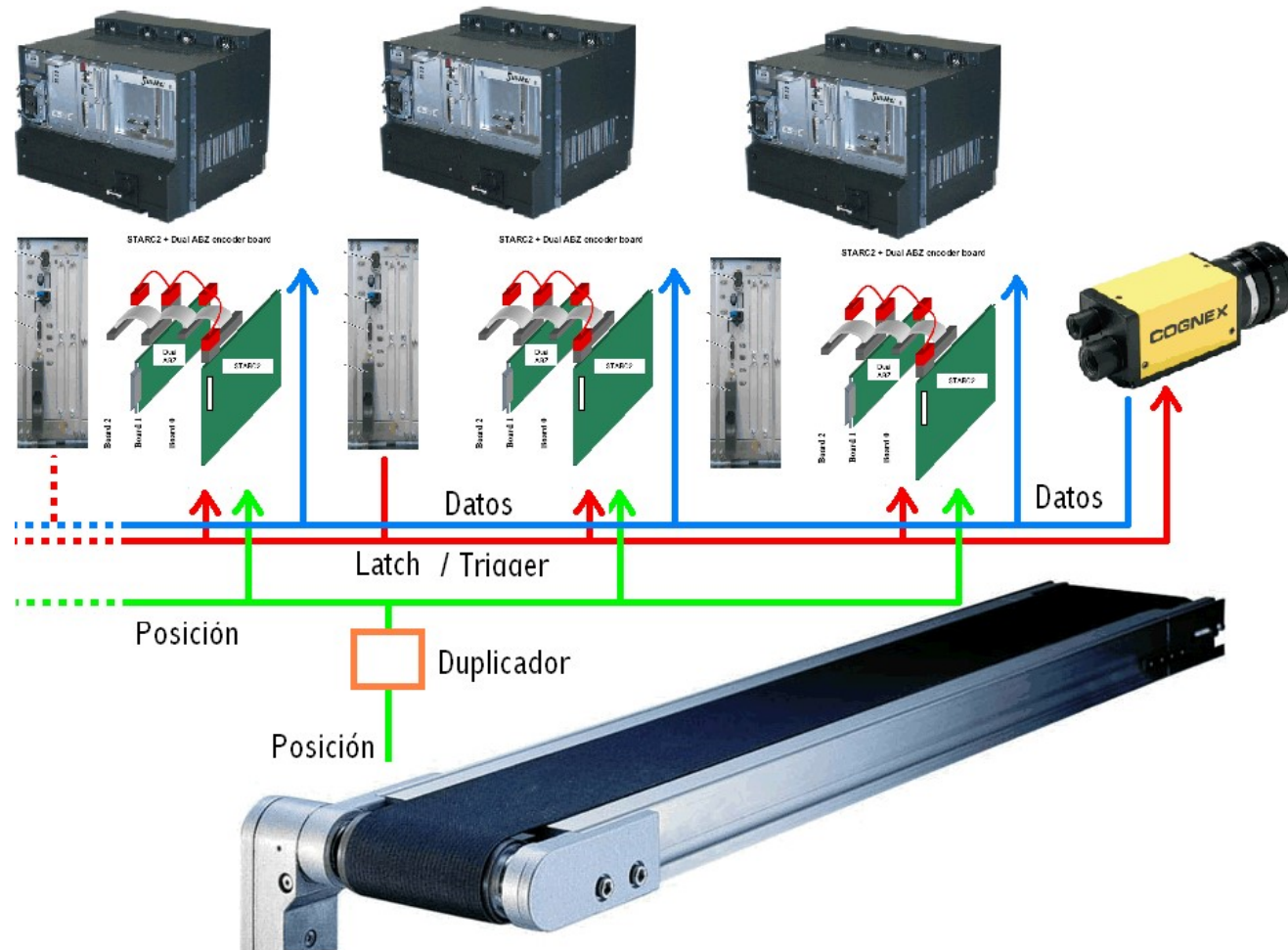
CONFIGURACIÓN 1 CS8C Y VISIÓN



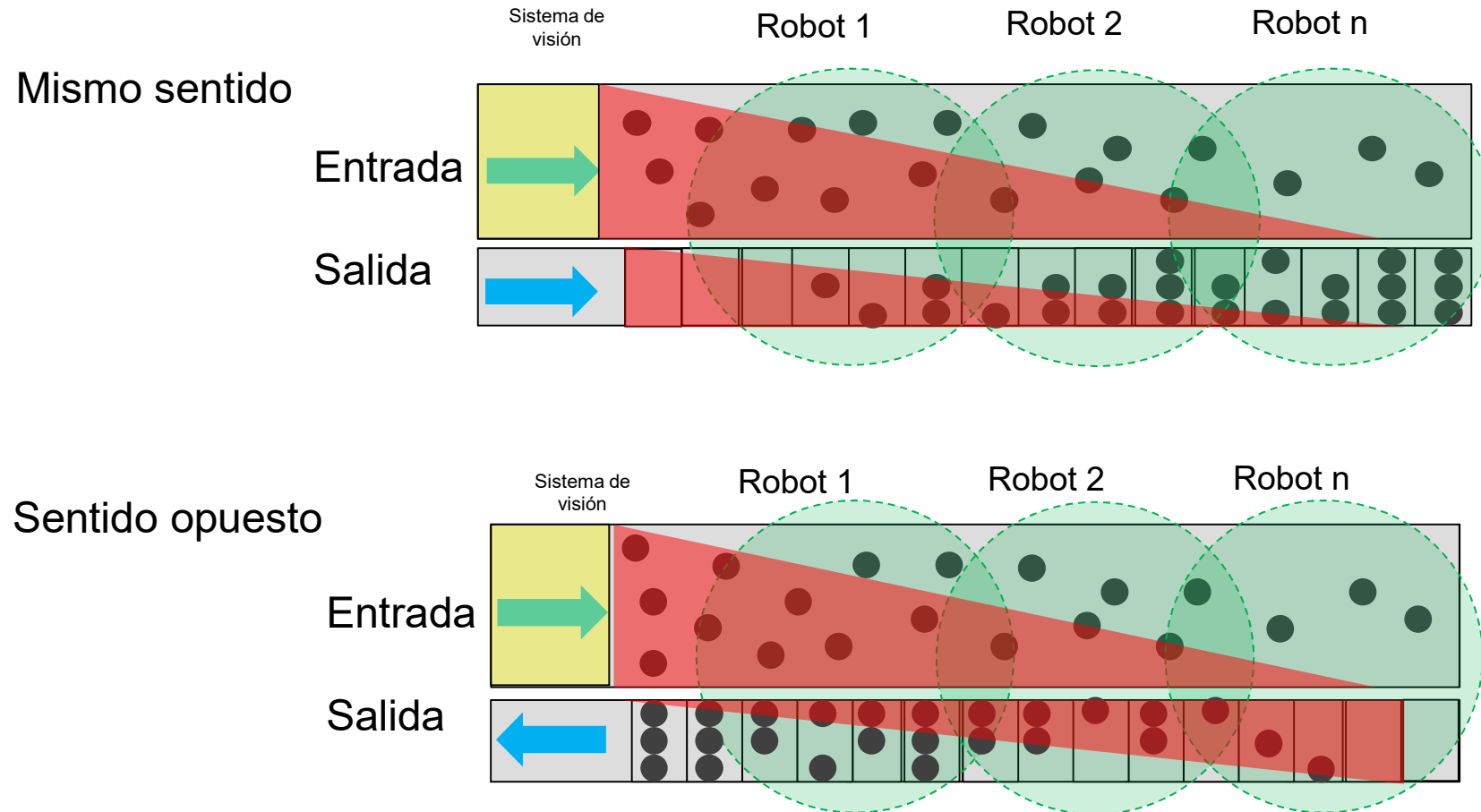
CONFIGURACIÓN 2 CS8C Y VISIÓN



CONFIGURACIÓN 2 CS8C Y VISIÓN



CONFIGURACIÓN 2 CINTAS



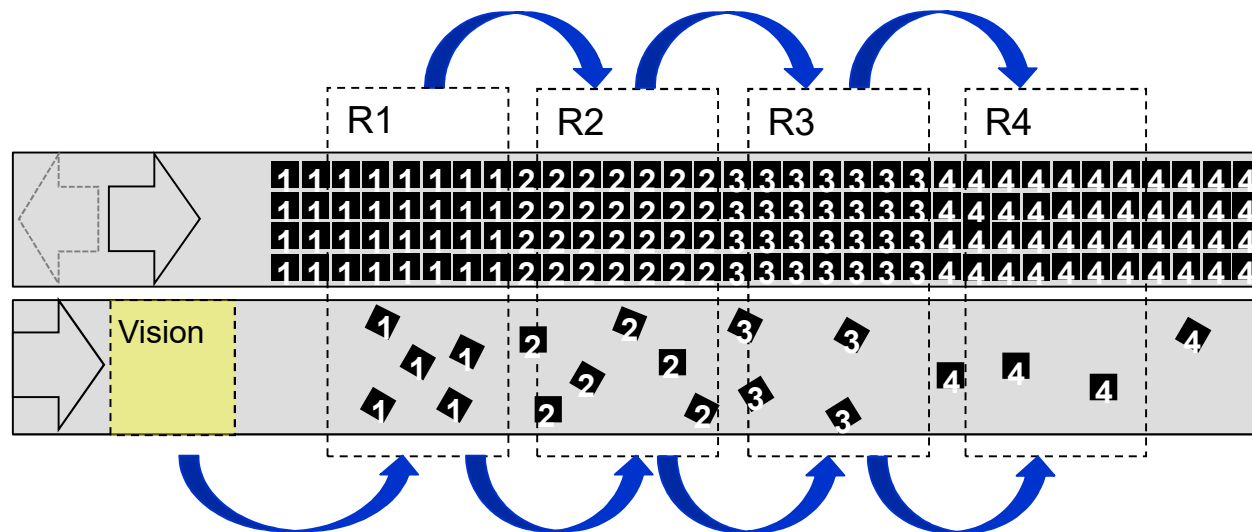
FORMACIÓN VAL3



DISTRIBUCIÓN

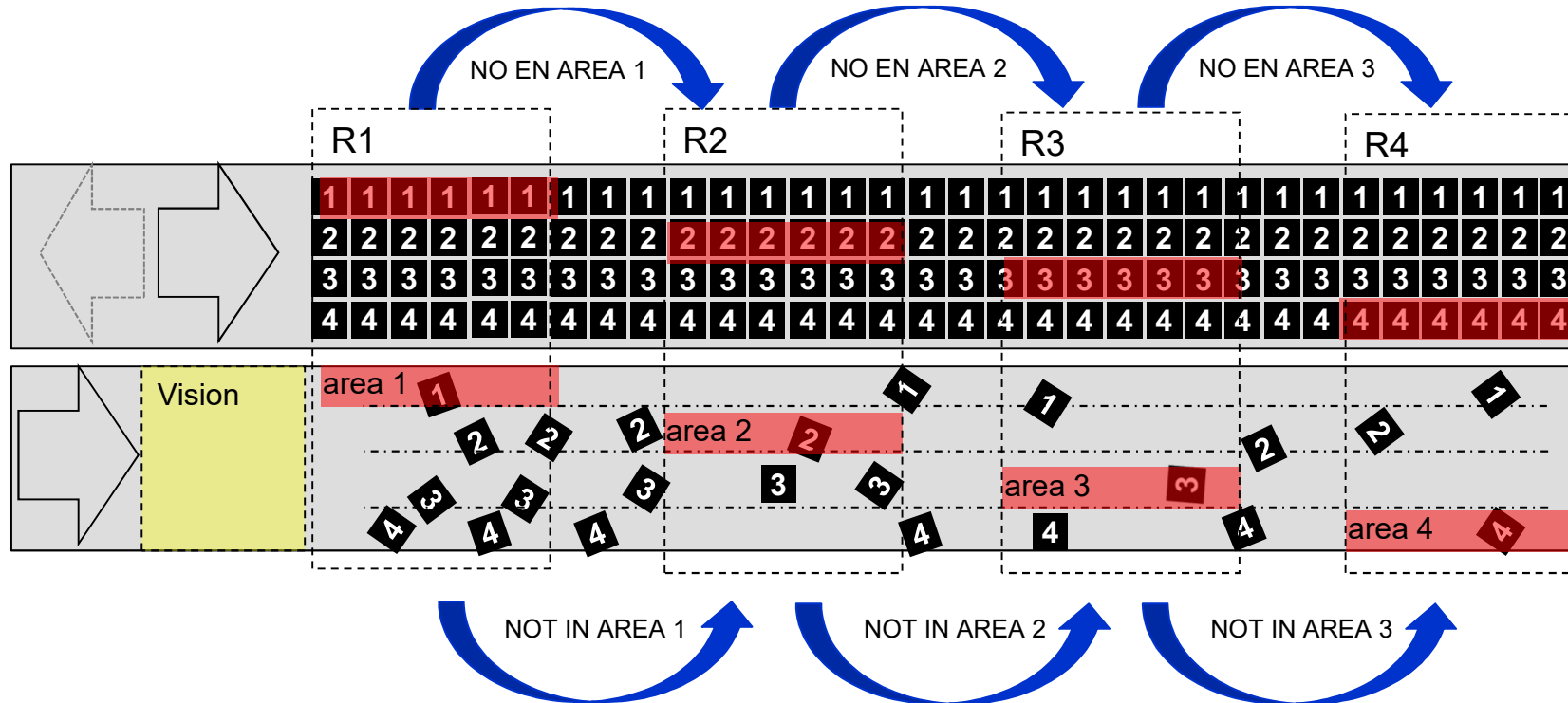


POR DESBORDAMIENTO



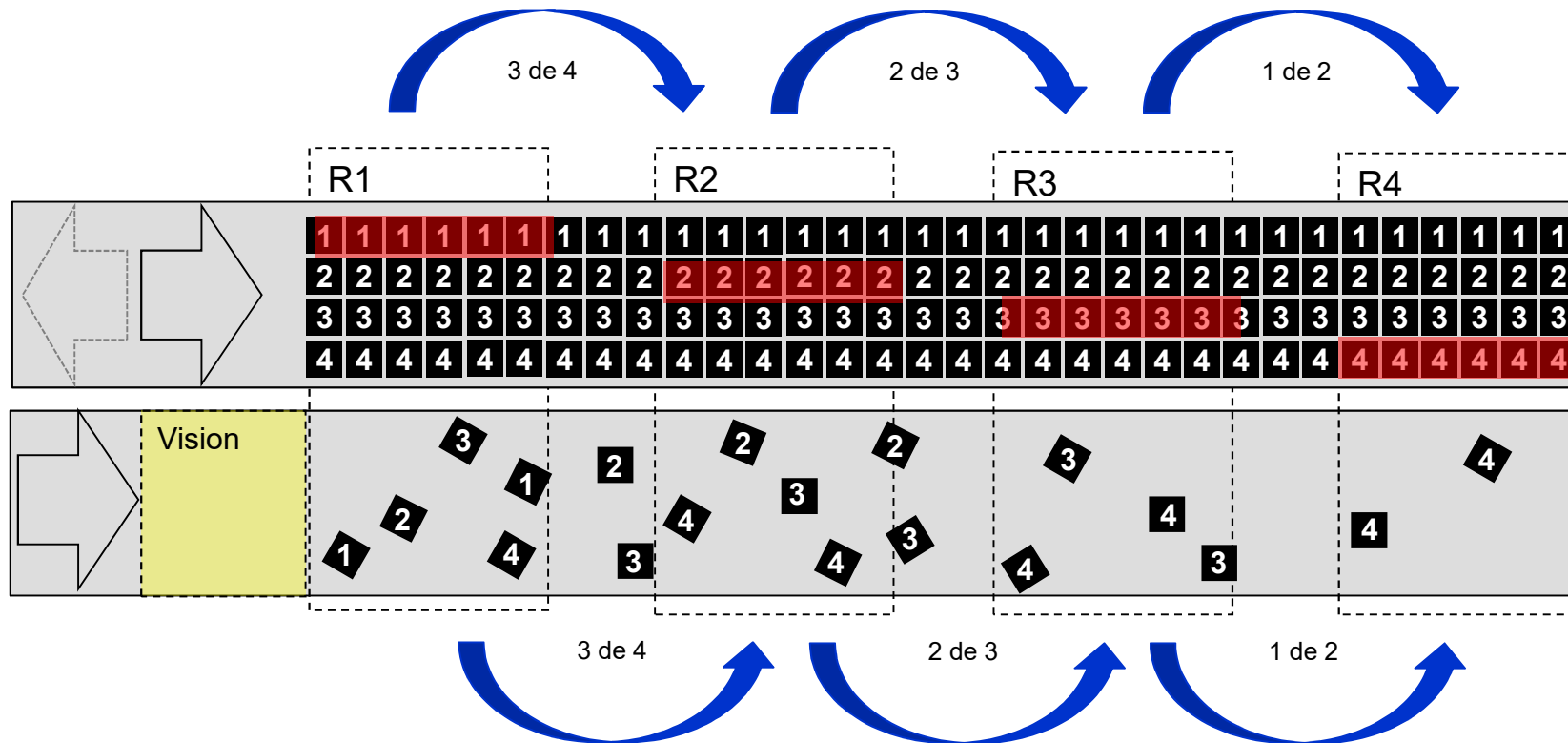
- Es la mejor configuración para gestionar la variación de flujo en la cinta de estrada
- El flujo de piezas no se distribuye por igual entre los robots.
- Es posible que el último robot se quede sin partes para llenar las cavidades de salida.

POR AREA



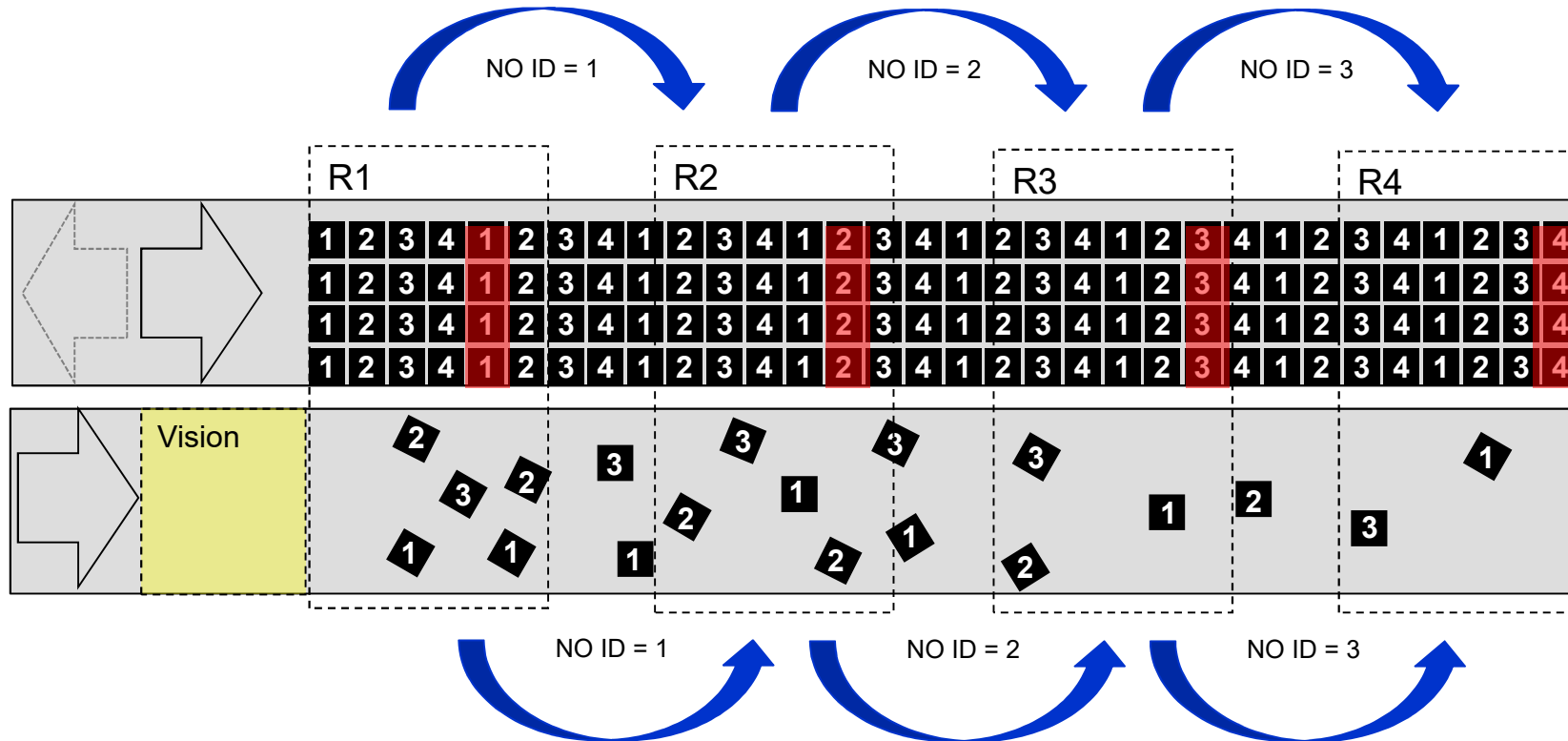
- Permitir asignar diferentes áreas de la cinta transportadora para cada robot.
- Distribución de la carga entre los robots.

POR DISTRIBUCIÓN



- Distribución de la carga entre los robots.

POR IDENTIFICADOR

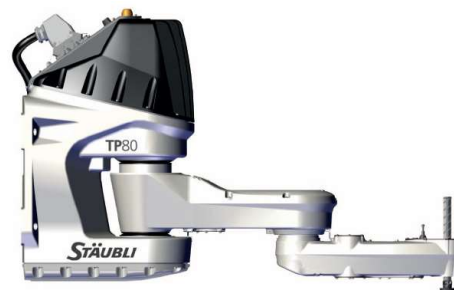


- Distribución de la carga entre los robots.
- Se asigna el de tipo de pieza por el robot.
- Coincidencia entre la recogida y colocación de ubicación.
- Permite la gestión de las piezas defectuosas.

FORMACIÓN VAL3



SEÑALES



SEÑALES

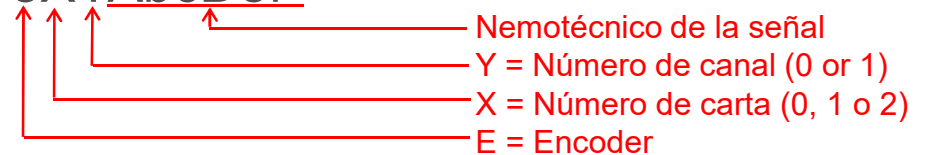
Entradas salidas digitales.

```
10%
==Control panel==
-Dinputs
  e00LatchSig = Off
  e00Latch = Off
  e00PowerErr = Off
  e00HwErr = Off
  e00OvsErr = Off
  e00CountsErr = Off
-Doutputs
  e00EnLatch = Off
  e00EnPrst = Off
  e00RstErr = Off
  e00LatchEdgFall = Off
-Rinputs
```

Entradas salidas analógicas.

```
10%
==Control panel==
-Rinputs
  e00CurrPos = 0
  e00LatchPos = ?
  e00CountsMes = 4096
-Routputs
  e00Counts = 4096
  e00PrstPos = 0
  e00LatchFilter = 0
+EncoderIO-B0_1
+EncoderIO-B1_0
+EncoderIO-B1_1
+EncoderIO-B2_0
+EncoderIO-B2_1
```

Standard nomenclatura señales: eXYAbcDef



Entradas Digitales

- e00LatchSig = Señal de trigger (disparo). Entrada en la tarjeta dual ABZ.
- e00Latch = Se ha leído una señal de trigger y la posición del encoder ha sido capturada.
- e00PowerErr: Es cuando el fusible de la tarjeta ABZ esta fundido.
- e00HwErr: Cuando existe un problema en el cableado. Un cable esta roto, no esta bien (ejemplo A → B, B → A), algún problema en los cables.
- e00OvsErr: Sobre velocidad. Cuando el encoder avanza mas de 180 en un ciclo de sistema (4 o 2 ms dependiendo del robot)
- e00CountsErr: Cuando e00CountsMes y e00Counts no son iguales.

Salidas digitales

- e00EnLacht: Esta salida permite captura la posición del encoder cuando se activa el trigger.
 - Se activada desde una tarea del Val3.
 - El sistema la desactiva tras activar la señal e00Latch .
- e00EnPrst: Sirve para poner un determinado valor en la entrada e00CurrPos del encoder.

- e00RstErr: Reinicializa las variables de error.
- e00LatchEdgFall: Esta señal permite configurar el disparo (trigger) por flanco ascendente (false) o descendente (true).

Entradas analogicas

- e00CurrPos: Posición actual del encoder.
- e00LatchPos: Posición capturada del encoder tras una señal de trigger:
 - Si e00EnLatch=off, esta señal es igual a ???.
 - Cuando e00Latch=on, posición del encoder en el momento del trigger.
- e00CountsMes: Número de pulsos por vuelta del encoder medido por el sistema.

Salidas analogicas

- e00Counts: Número de pulsos por vuelta del encoder definido en el programador.
- e00PrstPoss: El valor a poner en e00CurrPos cuando se activa la señal e00EnPrst.
- e00LatchFilter: Tiempo que debe permanecer activada e00LatchSig para considerarla como un verdadero trigger (no tocar).

FORMACIÓN VAL3

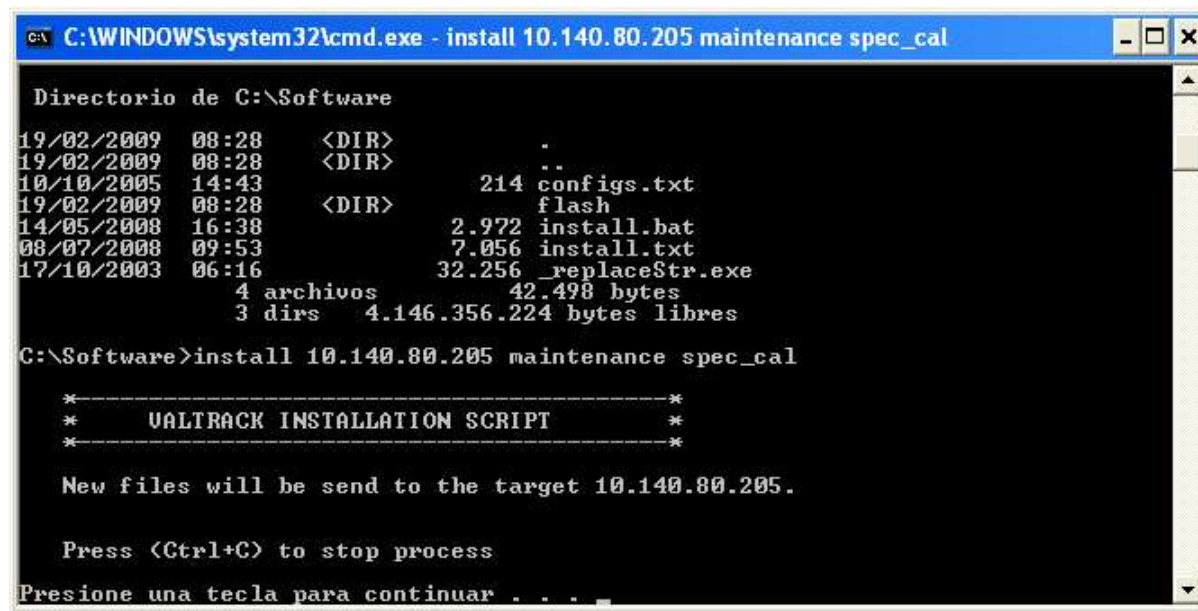


PRIMEROS PASOS



INSTALACIÓN DEL SOFTWARE

- Copiar la carpeta Software que acompaña al software en la raíz de nuestro PC.
- En windows: Inicio → Ejecutar → Cmd
- Ejecutar: cd c:\software
- Ejecutar: install IP_controlador login password (ejemplo: install 10.140.80.205 maintenance spec_cal).



```
C:\WINDOWS\system32\cmd.exe - install 10.140.80.205 maintenance spec_cal

Directorio de C:\Software
19/02/2009  08:28    <DIR>          .
19/02/2009  08:28    <DIR>          ..
10/10/2005  14:43                214 configs.txt
19/02/2009  08:28    <DIR>          flash
14/05/2008  16:38                2.972 install.bat
08/07/2008  09:53                7.056 install.txt
17/10/2003  06:16            32.256 _replaceStr.exe
                4 archivos          42.498 bytes
                3 dirs    4.146.356.224 bytes libres

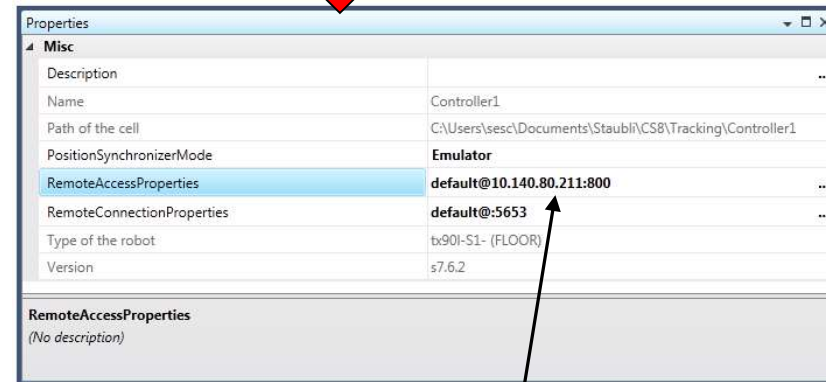
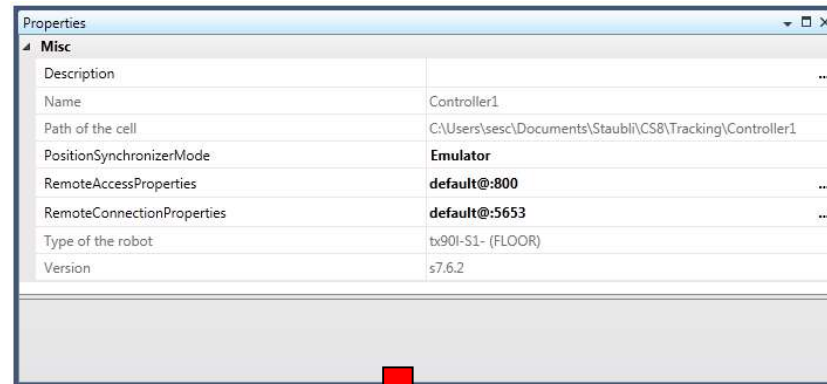
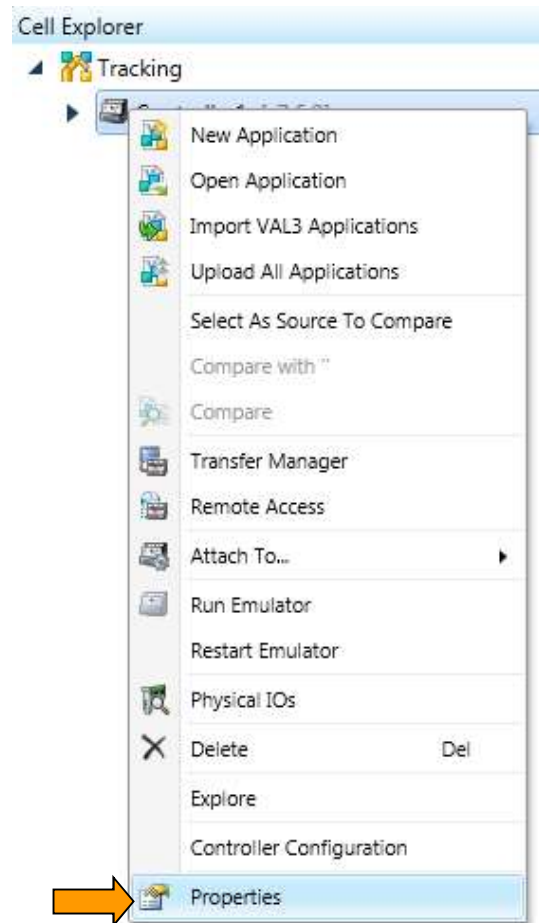
C:\Software>install 10.140.80.205 maintenance spec_cal

*-----*
*   VALTRACK INSTALLATION SCRIPT   *
*-----*

New files will be send to the target 10.140.80.205.

Press <Ctrl+C> to stop process
Presione una tecla para continuar . . .
```

LICENCIAS



Ip del controlador

Target Properties

IP: 10.140.80.211

Port: 800

User name: default

Password:

Comment:

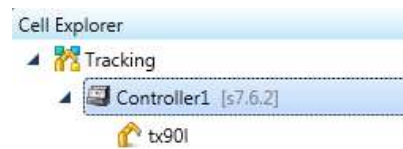
OK Cancel



Properties

| | |
|----------------------------|--|
| Misc | |
| Description | ... |
| Name | Controller1 |
| Path of the cell | C:\Users\sest\Documents\Staubli\CS8\Tracking\Controller1 |
| PositionSynchronizerMode | Emulator |
| RemoteAccessProperties | default@10.140.80.211:800 |
| RemoteConnectionProperties | default@10.140.80.211:5653 |
| Type of the robot | tx90I-S1- (FLOOR) |
| Version | s7.6.2 |

Misc
(No description)



| Option | Value |
|--------------------|-------|
| remoteMcp | None |
| testMode | None |
| mcpMode | None |
| lli | None |
| PaintiXen | None |
| valPlast | None |
| valTrack | None |
| valTending | None |
| valKernel | None |
| valHSM | None |
| alter | None |
| oemLicence | None |
| remoteMaintenance | None |
| pciAccess | None |
| 6axesAbsoluteRobot | None |
| 4axesAbsoluteRobot | None |
| continuousAxis | None |
| advCtrlFunctions | None |
| spline | None |
| carriedPart | None |

Disabled Demo Activated ✕
..... Check

- Disabled: sin licencia.
- Demo: El software funciona durante dos hora y luego hay que parar y encender el controlador para disponer de dos hora más.
- Activated: Con licencia y pleno funcionamiento, para lo cual tendremos que introducir la licencia.

Una vez parado y arrancado el controlador podremos comprobar si tenemos la licencia en: tecla menú → panel de control → configuración del controlador → versiones

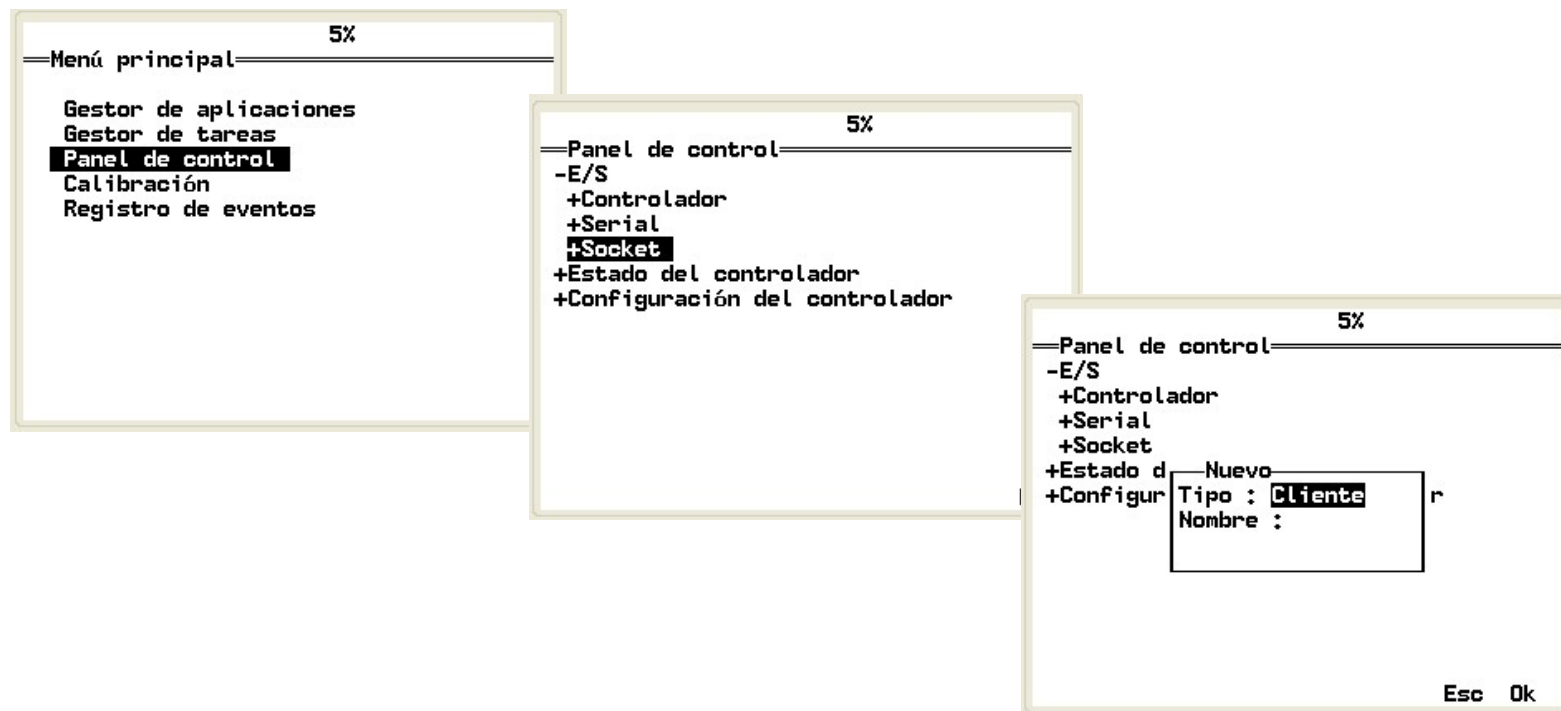
```
50%
=====Panel de control=====
-Versiones
  System: s7.6.1 - Jul 24 2013
  Configuration Version: c1.017
  CPU: EM07N
  Option testMode: enabled
  Option valPlast: demo
  Option valTrack: demo
  Option valTending: demo
  Option VAL 3: enabled
  N/S del controlador: : F10_5FQ3A1_C_
  Brazo: tx60L-S1-R2 - suelo
  N/S del brazo:: F10_5FQ3A1_A_01
  Nombre del robot:
```



COMUNICACIÓN CON COGNEX

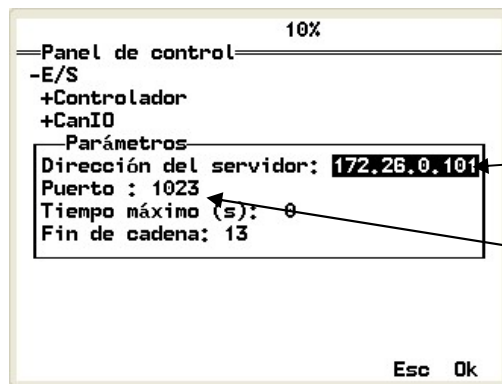
Para que el ValTrack pueda comunicarse con una cámara Cognex hace falta crear un socket de comunicación en el controlador.

El nombre del Socket es cognex1 para la librería Cognex1 y cognex2 par la librería Cognex2.

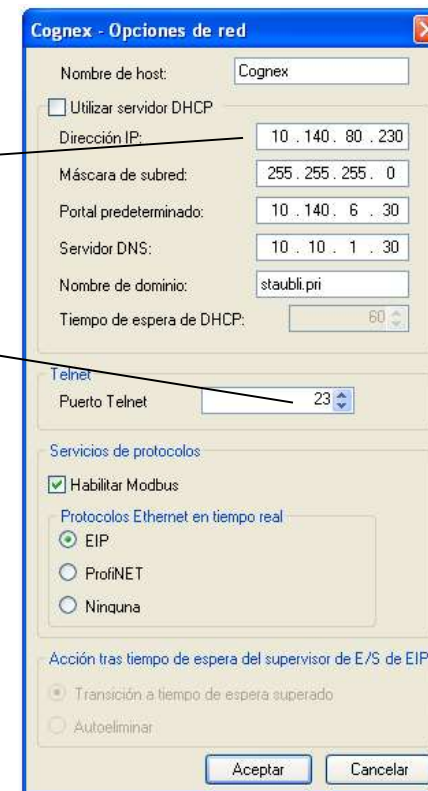


El puerto Telnet por defecto de la cámara es el 23, este puerto ha de coincidir con el del controlador.

Los puertos por debajo de 1000, son puertos reservados para determinadas funciones, por lo que es altamente recomendable utilizar puertos superiores a 1000 para evitar posibles interferencias.



Configuración Controlador



Configuración cámara

PUNTOS HARDWARE A VERIFICAR

- Cableado del encoder y sensor.



- Que el valor actual de encoder (exyCurrPos) cambia en el panel de control cuando la cinta se esta moviendo.
- El valor del encoder ha de incrementar cuando la cinta se mueva hacia adelante, si decrementa cruzar los pares A y B ($A \leftrightarrow B$, $/A \leftrightarrow /B$).
- La señal exyLatchEdgFall está correctamente definida.
- Simular un trigger y verificar que la posición del encoder es capturada.
- Introducir el valor de pulsos de encoder en la señal e00Counts.



FORMACIÓN VAL3

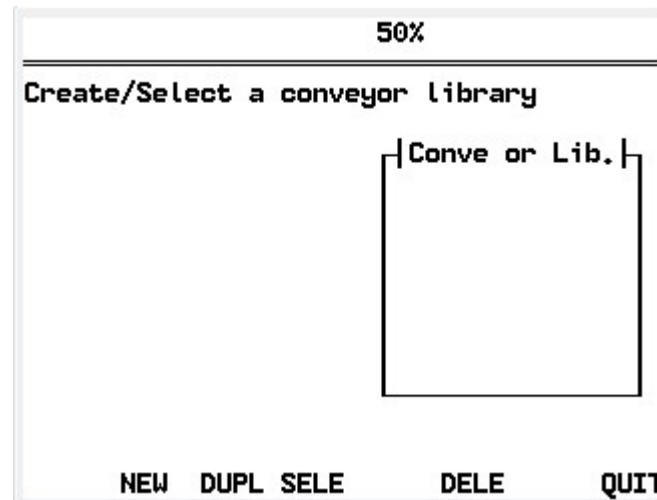


WIZARD



CALIBRACIÓN DE LA CINTA

Ejecutar el programa Wizard.



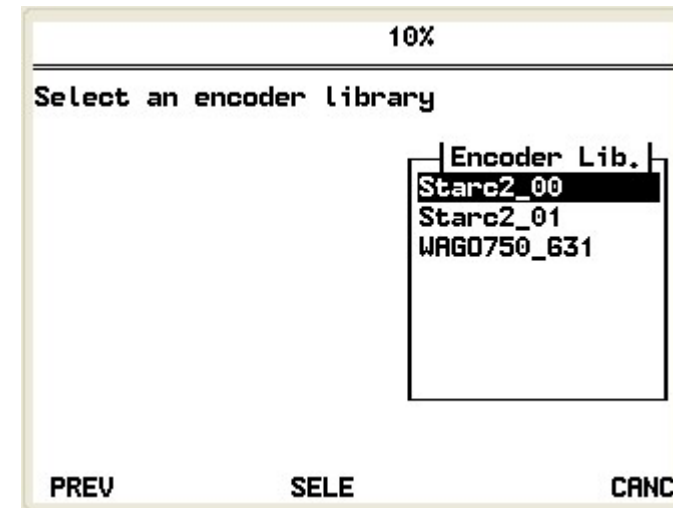
Ventana principal del Wizard, donde podremos seleccionar/crear una librería con la configuración de nuestro sistema.

Podremos tener distintas librerías según nuestras necesidades, por ejemplo:

- Sensor
- Cámara
-

Podremos escoger una serie tarjetas lectoras de encoder.

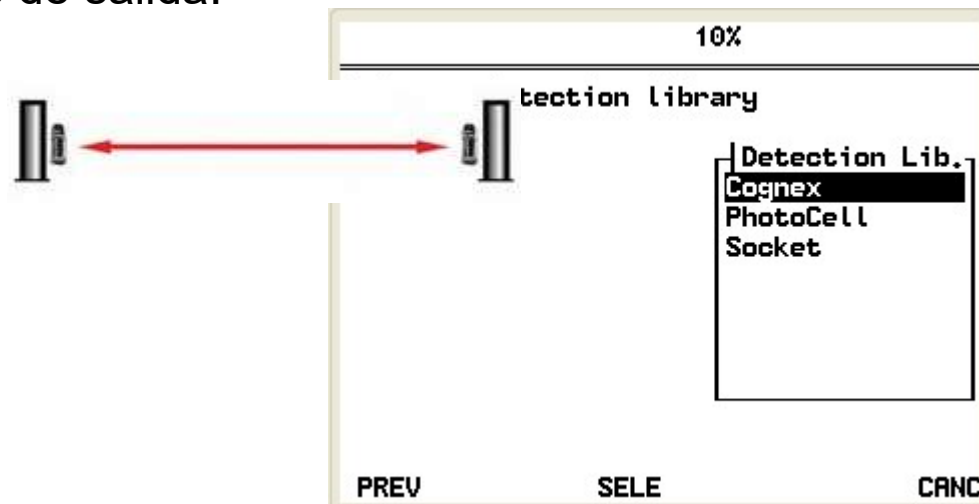
- Las tarjetas Dual ABZ aparecen bajo la nomenclatura Starc2_XY, donde X es la posición física de la tarjeta (0, 1 o 2) y Y el canal de la tarjeta (0 o 1)
- Wago750_G31 por Profibus.



Tenemos tres opciones de detectores:

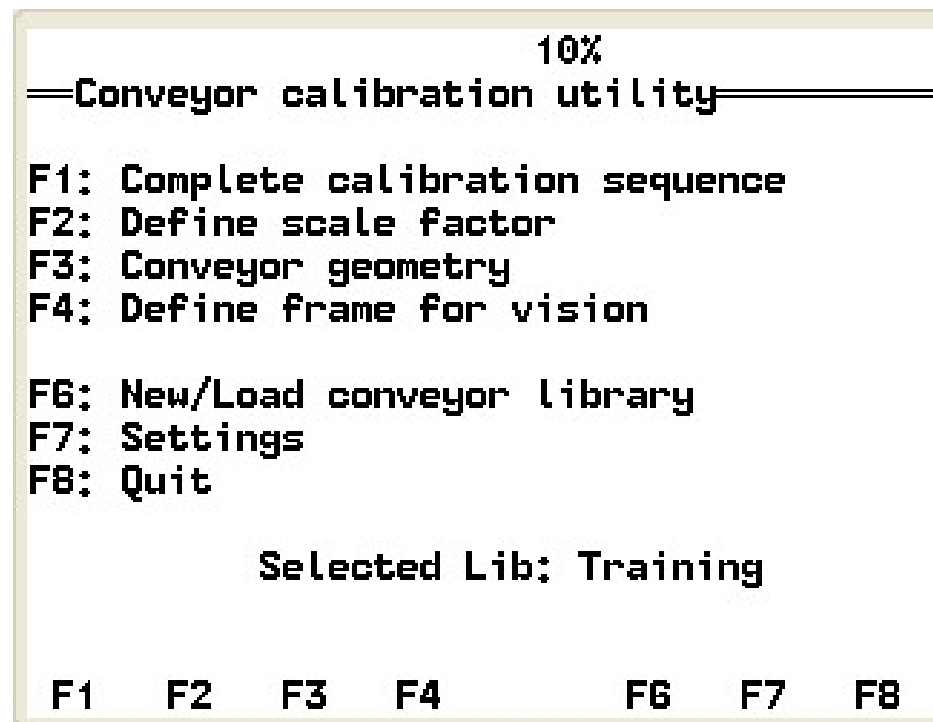
- Cognex1 y Cognex2: El sistema de detección de piezas es mediante una cámara Cognex (tenemos desplazamientos y rotación). Es necesario tener definido el socket antes de seleccionar esta opción.
- Sensor1 y Sensor2: Se detecta las piezas mediante un sensor.
- Socket1 y Socket2: Los datos de la detección vienen de un dispositivo previo (CS8C, PC, etc).

Es recomendable utilizar las librerías 1 para las cintas de entrada y las librerías 2 para las de salida.



CALIBRACIÓN DE LA CINTA

El programa Wizard nos permite configurar los parámetros de nuestro sistema.



F7: Parámetros generales

Calibrated Pointer: Características del puntero que se utilizará durante todo el proceso de calibración.

Software: Versión del ValTrack.

Event Logger: Si es diferente de cero se envía por telnet mensajes de error del software.

0. Nada

1. Detección

2. Movimiento

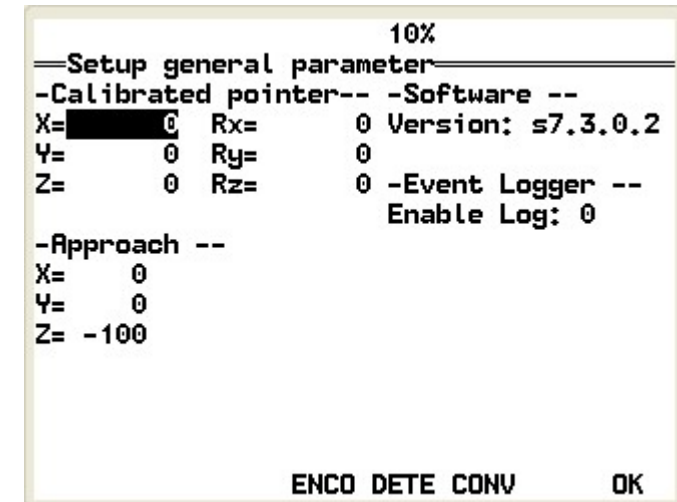
4. Cola de movimientos.

8. Varios

Es posible poner dos parámetros a la vez. (1 y 2 = 3, 4 y 2 = 6,...)

Es mejor poner este parámetro a cero cuando no estamos en depuración de programa (en producción).

Approach: Distancia de aproximación. Cuando queremos ir a una posición, se realiza una aproximación al punto donde estamos seguida de una aproximación a donde queremos ir, y finalmente vamos al punto.



F7(Parámetros generales) + F4: Parámetros del encoder

LatchPeriod: Número de trigger de detección de pieza que han de suceder para que se considere que se ha detectado una pieza. Normalmente vale 1.

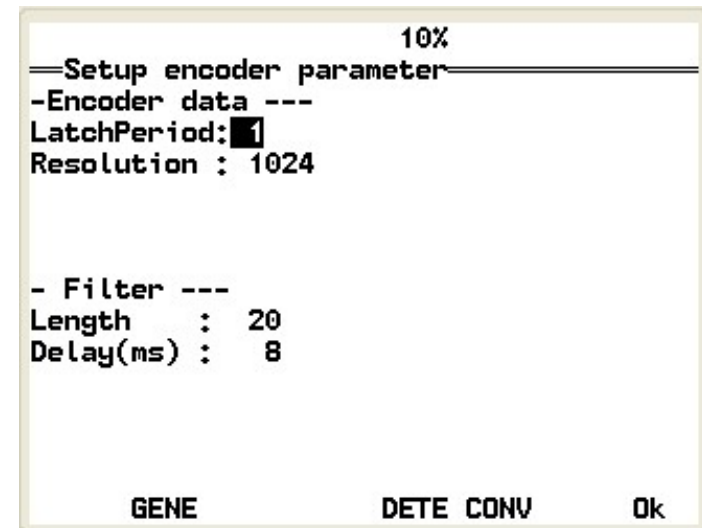
Resolution: Número de pulsos por revolución del encoder.

Filter: Parámetros de filtrado de la señal del encoder.

Lengh: Con los n valores anteriores se calcula la rampa de posición para estimar la siguiente posición y tener una rampa de velocidad / posición más estable.

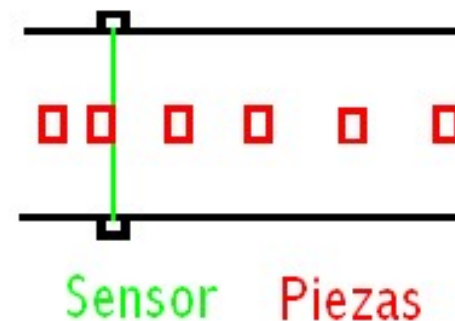
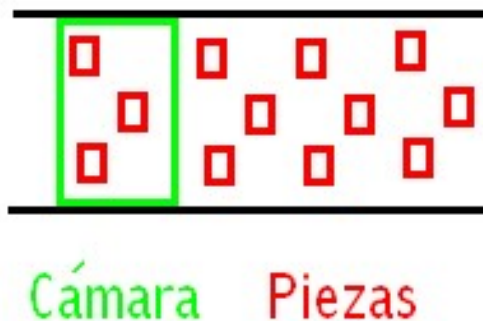
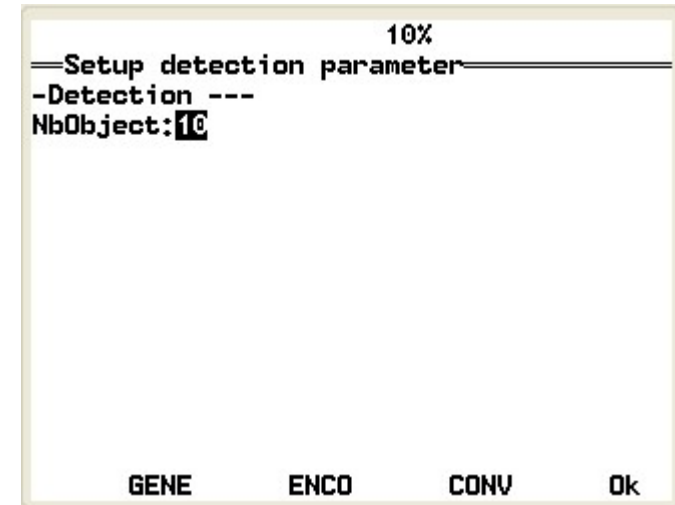
- Valor grande: movimiento más homogéneo.
- Valor pequeño: movimiento más inestable o a golpes, pero reacción más rápida a los cambios de velocidad.

Delay (ms). Retardo de comunicación del valor de encoder entre las diferentes capas de software y hardware. No editable.



F7(Parámetros generales) + F5: Parámetros de detección

NbObject: Número máximo de piezas que pueden ser introducidas en la pila de detección por cada trigger.

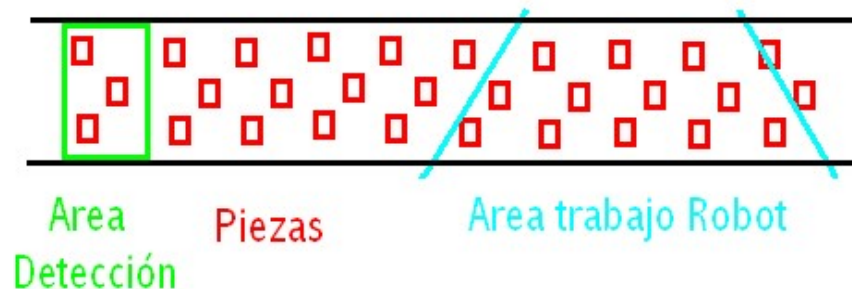
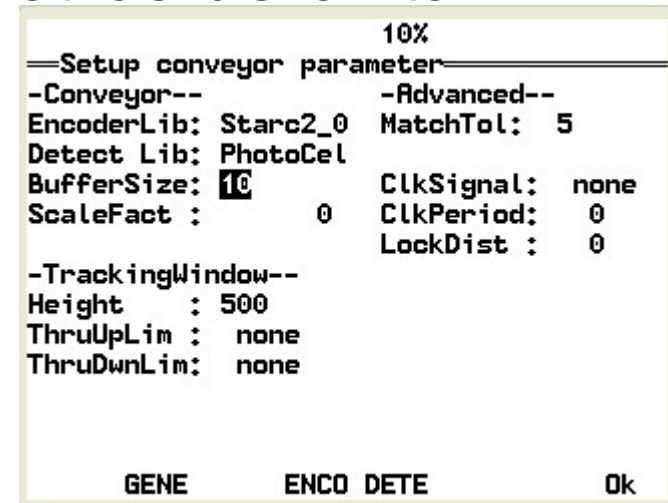


F7(Parámetros generales) + F6: Parámetros de cinta

Encoder Lib: Tarjeta lectora de encoder que utilizamos.

Detect Lib: Tipo de sistema de detección de pieza.

BufferSize: Número máximo de piezas que puede haber entre el área de detección y el área de trabajo del robot.



ScaleFact: Relación entre el desplazamiento de la cinta transportadora y los grados del encoder (mm/°). Se define en el apartado: “Define scale factor”.

MatchTol: Tolerancia en mm para saber que dos objetos son el mismo.

ClkSignal: ValTrack puede generar una señal de disparo cada x mm (definido en el campo ClkPeriod). En este campo ponemos la salida que queremos usar para generar esta señal.

ClkPeriod: Distancia en mm para generar la señal de disparo.

LockDist: Distancia mínima entre dos señales de entrada de disparo., sirve para descartar cualquier disparo que suceda antes de completar la distancia mínima después del primer disparo.

Height: Altura del área de trabajo del robot.

ThruUpLim: Salida que activaremos si el robot está antes del área de trabajo y está por debajo de la altura de trabajo.

ThruDwnLim: Salida que activaremos si el robot ha pasado el área de trabajo y está por debajo de la altura de trabajo.

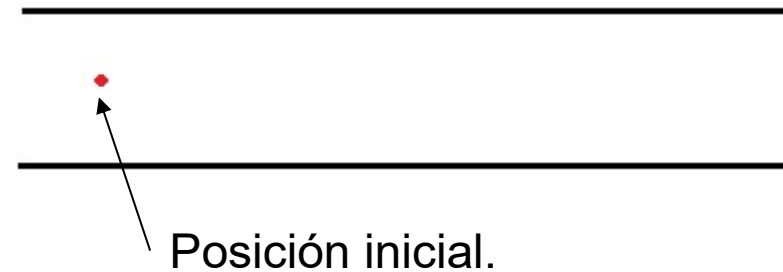
F2: Factor de escala.

```
50%
==Define scale factor 1/6==
Place calibration target on conveyor
as far upstream as the robot can reach

Jog the robot pointer on the target
and press NEXT to record the location

Robot position:
  X Y Z : 19.11   -401.14  -148.23
  RxRyRz: 0      -180      83.41
Encoder value : 0

NEXT CANCEL
```



```
1%
==Define scale factor 3/6==
Move the robot away from the target.

Run the conveyor until the target has
reached the downstream limit of the
robot's working range

Make sure the target doesn't slip on
the conveyor when it starts and stops

Press NEXT to continue

NEXT CANCEL
```

!!!Importante!!!: Referencia de calibración no se ha de mover por encima de la cinta durante este proceso.

```

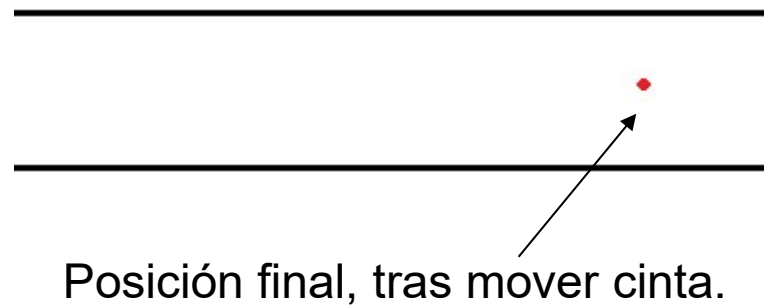
1%
==Define scale factor 3/6==
Without moving the calibration target
jog again the robot on the target

Press NEXT to record the location

Robot position:
  X Y Z : 639.81  -411.35  -212
  RxRyRz: 0      -180     83.41
Encoder value: 3719.74

PREV                                NEXT CANC

```



```

1%
==Define scale factor 4/6==
The encoder scale factor computation
gives the following result :

Encoder scale factor : 0.333

PREV                                NEXT CANC

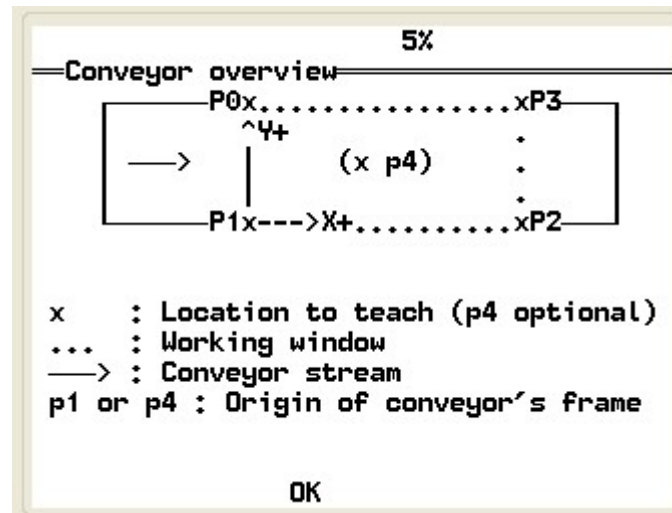
```

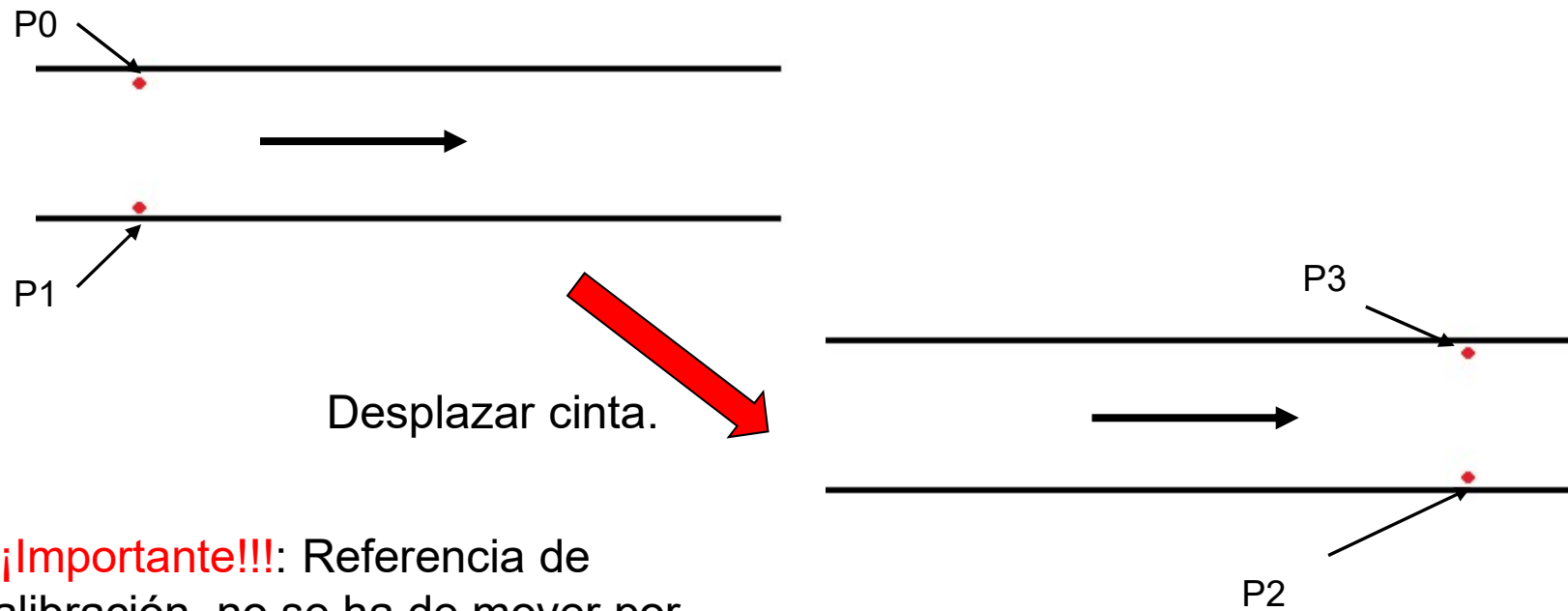
Relación milímetros cinta
y grados encoder.

F3: Geometría de la cinta

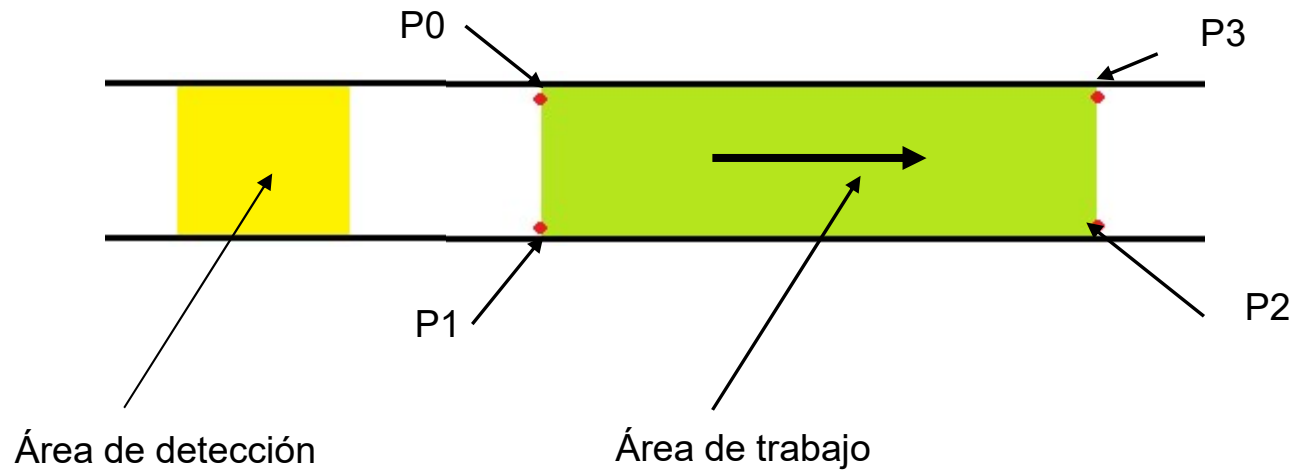
Mediante este procedimiento definiremos el área de trabajo del robot encima de la cinta.

Cuidado: con un robot scara y visión no se puede variar las alturas durante el proceso ya que implicaría rotaciones en RX y RY las cuales no se pueden realizar con un robot Scara. El plano de la cinta y el plano XY mundial del robot en tienen que ser paralelos.





¡¡¡Importante!!!: Referencia de calibración no se ha de mover por encima de la cinta durante este proceso.



F4: Plano de visión (sólo con cámaras)

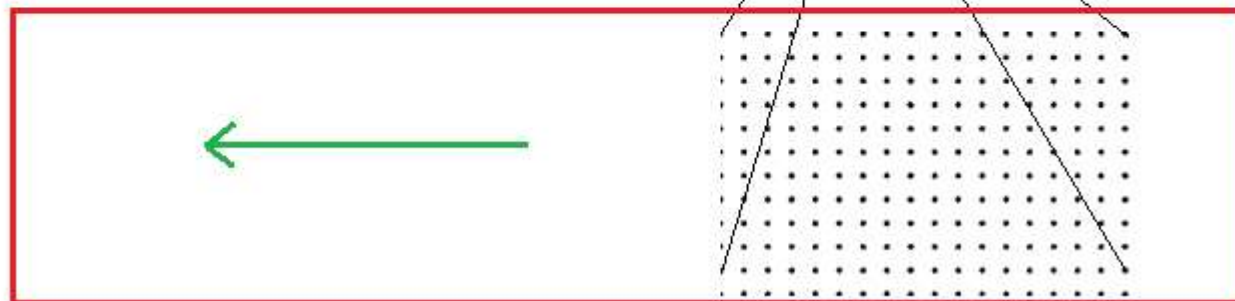
Información 25%

—Define vision frame—

- 1.-Place the calibration target on the conveyor, under the detection device.
- 2.-Trigger robot/vision latch.
Make sure that a picture is taken and that the Latched value is equal to the Encoder value (see below).
- 3.-Calibrate you vision system.
- 4.-Press NEXT to proceed

Encoder value : 50235.07
Latched value : 50235.07

LATC NEXT CANC



100%

Define vision frame

5.-Select one option:
 -Start the conveyor and stop it when the target is in the robot's working area
 ** Or **
 -Proceed if the target is already in the robot's working area.

6.-Press NEXT to proceed

Encoder value : 0
 Latched value : 0

PREV NEXT CANC

información 100%

Define vision frame

Teach the origin of the frame

Press HERE to store the location
 Press NEXT to continue

Current origin location
 X Y Z : 0 0 0
 RxRyRz: 0 0 0

PREV HERE MOVE

información 0.2%

Define vision frame

Teach the Xdirection of the frame

Press HERE to store the location
 Press NEXT to continue

Current Xdirection location
 X Y Z : 0 0 0
 RxRyRz: 0 0 0

PREV HERE MOVE

información 0.1%

Define vision frame

Teach the Ydirection of the frame

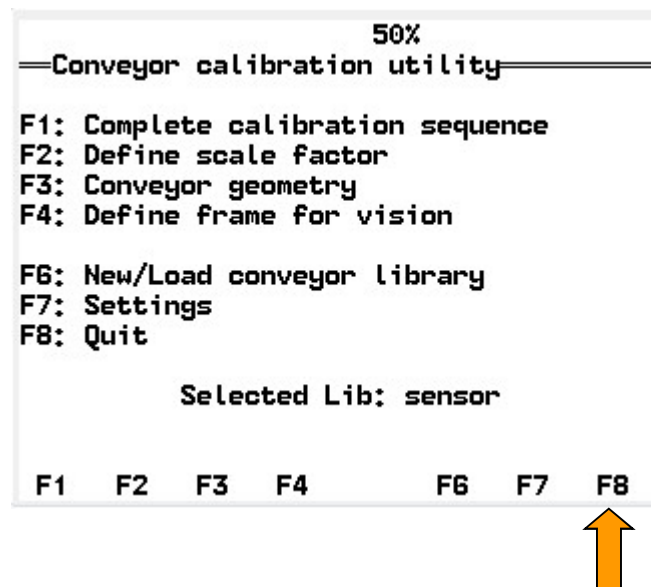
Press HERE to store the location
 Press NEXT to continue

Current Ydirection location:
 X Y Z : -104.96 -341.24 196
 RxRyRz: 180 0 170.31

PREV HERE MOVE NEXT CANC

Nota: Una vez finalizado el Wizard nunca salir parando el programa (tecla Stop del mando), utilizar siempre la tecla F8 de menú principal del Wizard.

Salir cerrando el programa no guarda las modificaciones de Wizard.



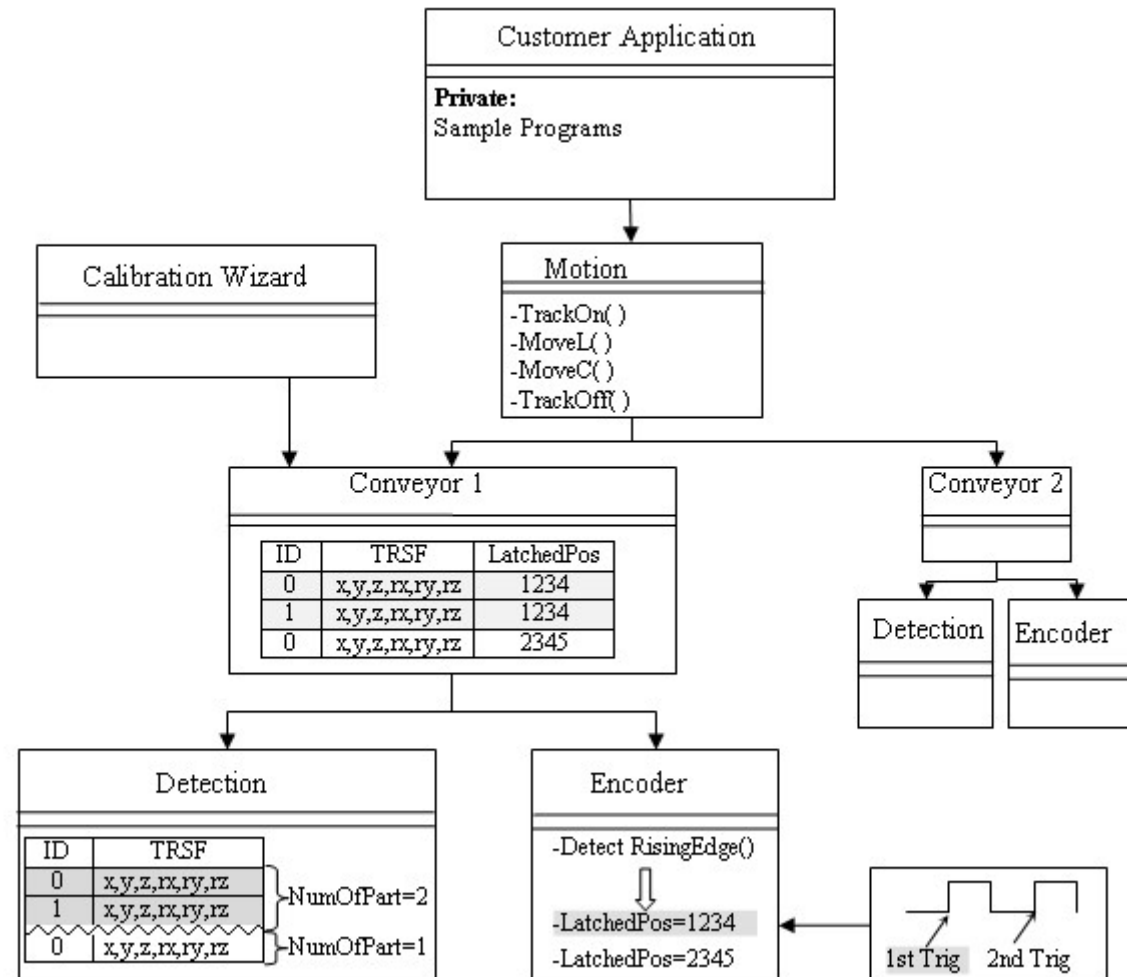
FORMACIÓN VAL3



PROGRAMACIÓN



RELACIÓN LIBRERIAS

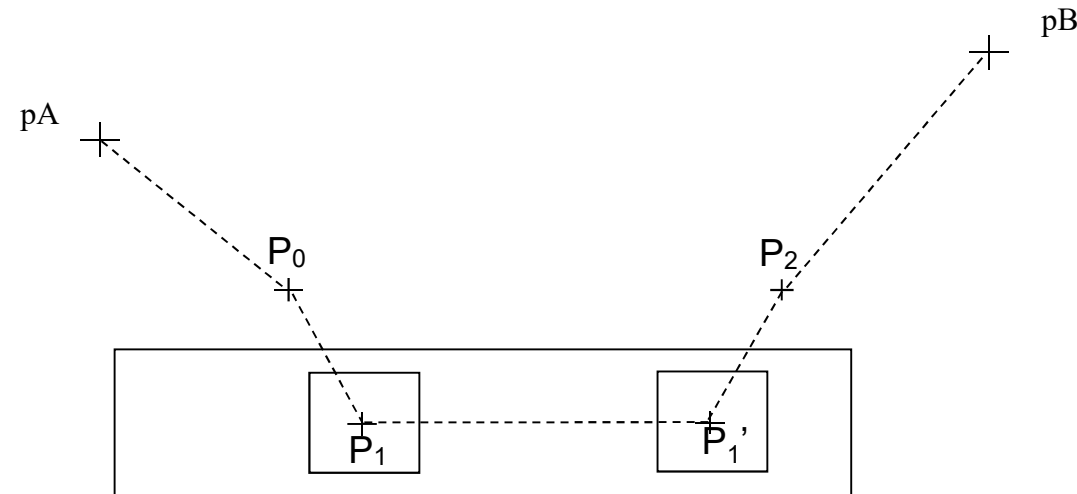


LIBRERÍA MOTION

La librería Motion proporciona a la aplicación del usuario, las funciones para generar movimientos sincronizados con las piezas móviles.

Las 3 funciones principales de la librería son:

- Sincronización con la pieza (pA-Po).
- Movimiento sincronizado (Po -P1-P1'-p2).
- Desincronización con la pieza (p2-pB).

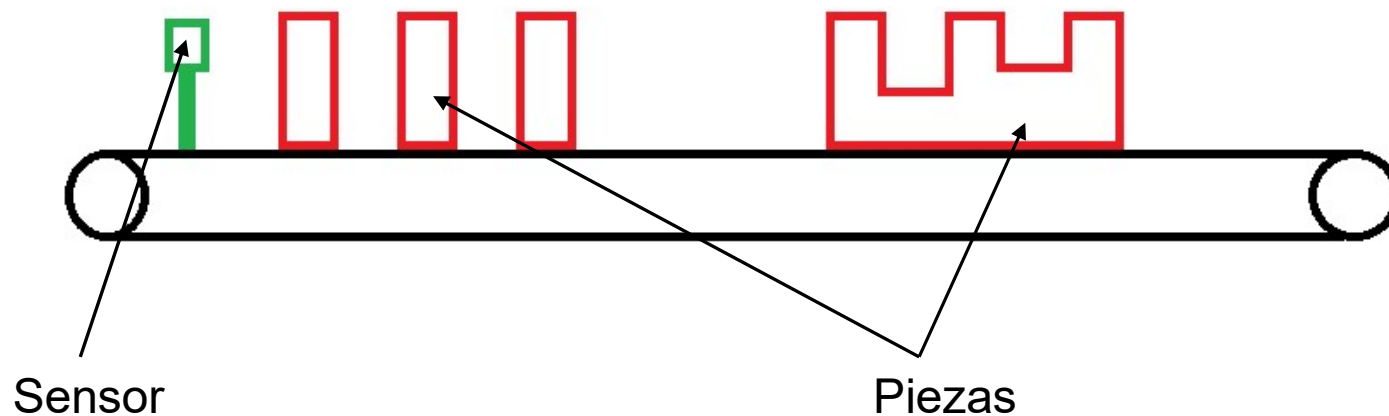


LIBRERÍA ENCODER

Es la librería que gestiona la información del encoder.

Realiza principalmente dos funciones:

- Activa y desactiva la detección de piezas sobre la cinta, pudiendo ser la detección por:
 - Distancia.
 - Por número de disparos.
- Actualizar las variables de ValTrack de:
 - Posición actual del encoder.
 - Posición de disparo (“Latch”)

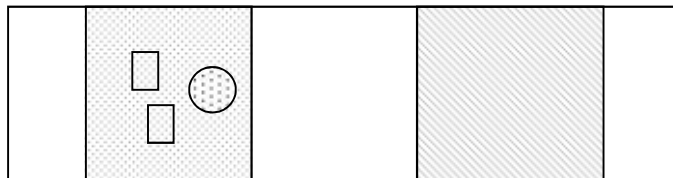


LIBRERÍA DETECTION

Es la librería que gestiona los dispositivos de detección.

Esta librería proporciona la siguiente información a la librería Conveyor dependiendo del tipo de detección:

- Detección simple (sensor):
 - Pieza detectada.
 - Tipo de pieza detectada (Object ID).
- Detección compleja (Visión)
 - Objeto detectado.
 - Número de objetos detectados a la vez.
 - Tipos de objetos detectados (Object ID).
 - Posición de los objetos.



Ventana de detección

Ventana de Tracking

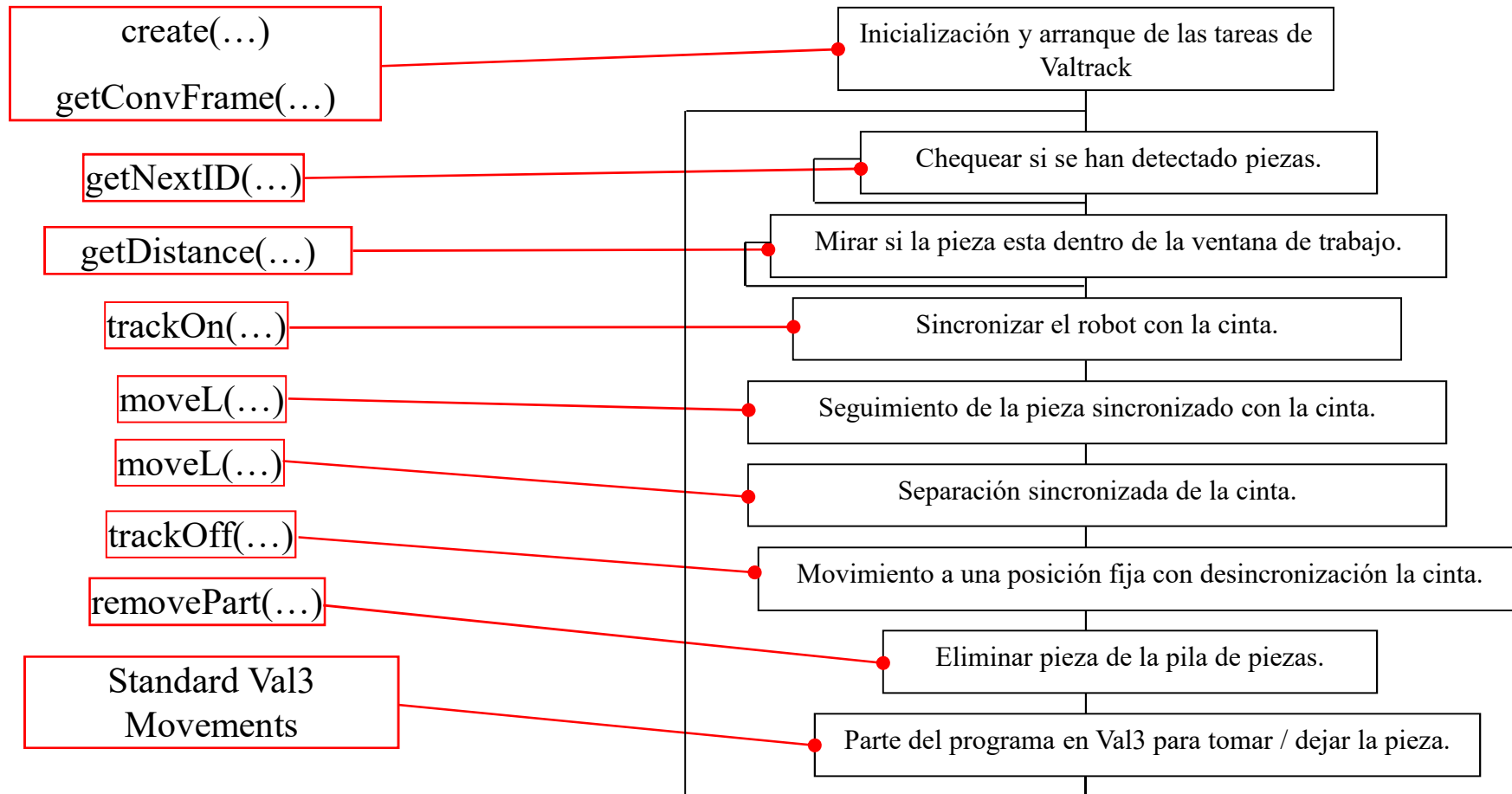
Array de transformadas

| |
|---|
| $\{X_{sq1}, Y_{sq1}, Z_{sq1}, RX_{sq1}, RY_{sq1}, RZ_{sq1}\}$ |
| $\{X_{sq2}, Y_{sq2}, Z_{sq2}, RX_{sq2}, RY_{sq2}, RZ_{sq2}\}$ |
| $\{X_{ci1}, Y_{ci1}, Z_{ci1}, RX_{ci1}, RY_{ci1}, RZ_{ci1}\}$ |

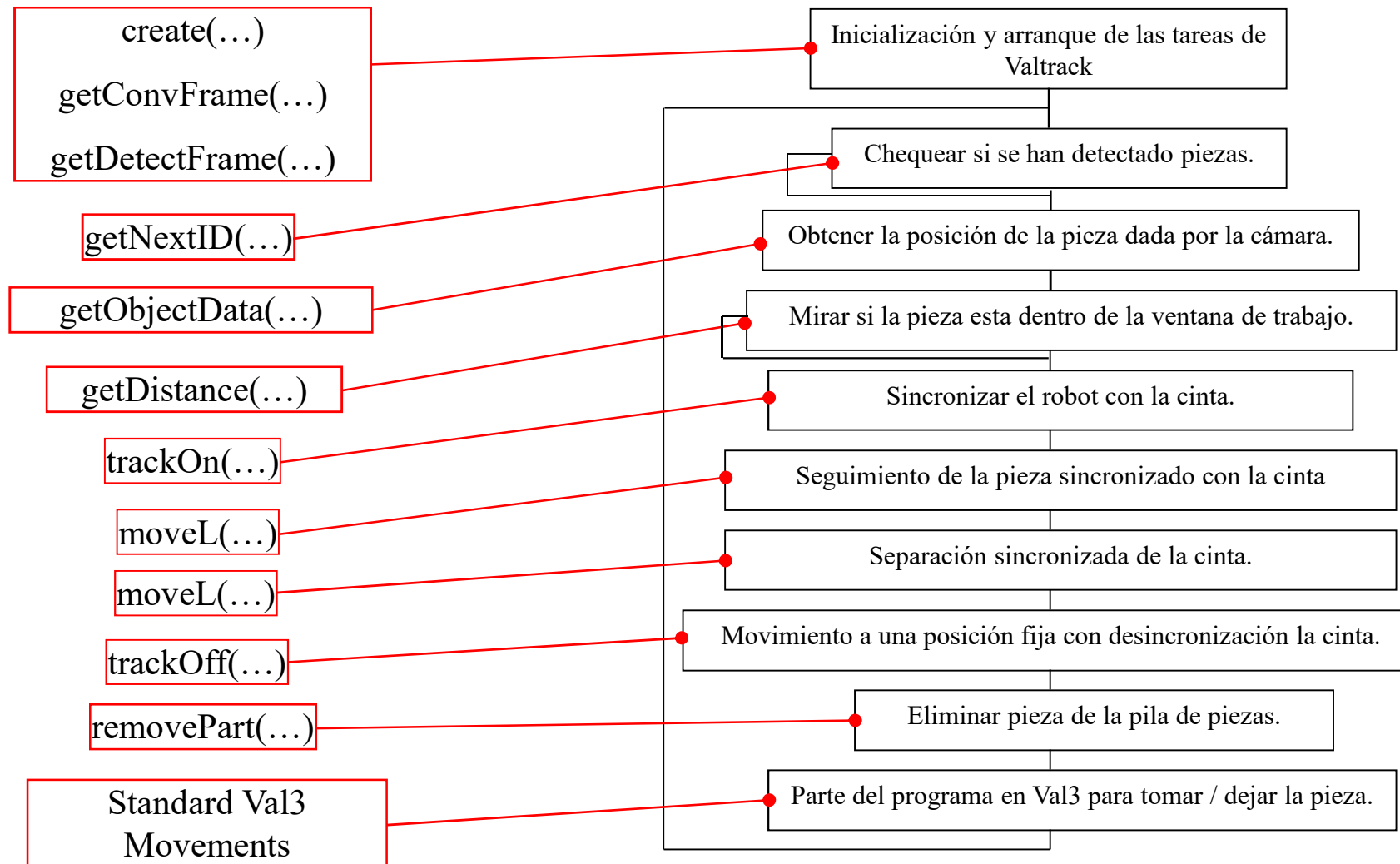
Array de IDs

| |
|---|
| 1 |
| 1 |
| 2 |

SECUENCIA PARA PIEZA



SECUENCIA PARA PIEZA (VISION)



PROGRAMAS LIBRERÍA MOTION

Arranque e inicialización del Valtrack:

- `create(sConveyorList)`
Lanza las tareas de Valtrack, donde SConveyorList es la lista de cintas transportadoras que se utilizaran.
- `getConvFrame(nCnvNb,fConveyor,fReference,l_nError)`
Devuelve la posición y orientación del plano de la cinta en referencia al plano fReference (por defecto será el plano mundial del robot) y el número de cinta transportadora nCnvN.
- `getDetectFrame(nCnvNb,fDetection,world,l_nError)`
Devuelve la posición y orientación del plano de la detección en referencia al plano fReference (por defecto será el plano mundial del robot) y el número de cinta transportadora nCnvN.

Enseñar/aprender una posición de tomar/dejar pieza (con sensor)

- `teach(nCnvNb,tGripper,fConveyor,pPick,l_nError)`
Devuelve el punto pPick en referencia al número de cinta transportadora nCnvNb y teniendo en cuenta la distancia desde el dispositivo de detección y la ventana de seguimiento.

Gestión de la cola de la cinta que contiene la lista de piezas detectadas:

- getNextID(nCnvNb,nID)
Devuelve el ID del primer objeto de la cola de la cinta transportadora nCnvNb. Un valor negativo significa que no hay piezas en la cola.
- getObjectData(nCnvNb,pPick.trsf,l_nError)
Devuelve las coordenadas de la pieza actual de la cola de la cinta transportadora nCnvNb. Es una orden para sistemas con visión.
- removePart(nCnvNb)
Elimina la pieza actual de la cola de la cinta transportadora nCnvNb.

Funciones para seguir pieza y generar movimientos de seguimiento:

- getDistance(nCnvNb,pPick,nTime,nDistanceToUp,nDistanceToDown,l_nError)
Devuelve la distancia la posición de la pieza actual de la cola de la cinta transportadora nCnvNb al limite inferior y superior del área de trabajo. Un valor negativo indica que no se ha llegado al limite y un valor positivo que se ha superado el limite.
- trackOn(nCnvNb,pApproPick,tGripper,mTrackOnOff,l_nMoveID,l_nError)
Movimiento de sincronización al punto pApproPick, normalmente será una aproximación al punto tomar / dejar.
- moveL(pPick,tGripper,mTracking,l_nMoveID,l_nError)
Movimiento linear sincronizado con el punto pPick (la pieza).

- moveC(plnter, pPick,tGripper,mTracking,l_nMoveID,l_nError)
Movimiento circular sincronizado con el punto pPick (la pieza).
- trackOff(pApproPlace,tGripper,mTrackOnOff,l_nMoveID,l_nError)
Movimiento de desincronización al punto pApproPlace.
- trackOffJ(jApproPlace,tGripper,mTrackOnOff,l_nMoveID,l_nError)
Movimiento de desincronización a la posición angular jApproPlace.

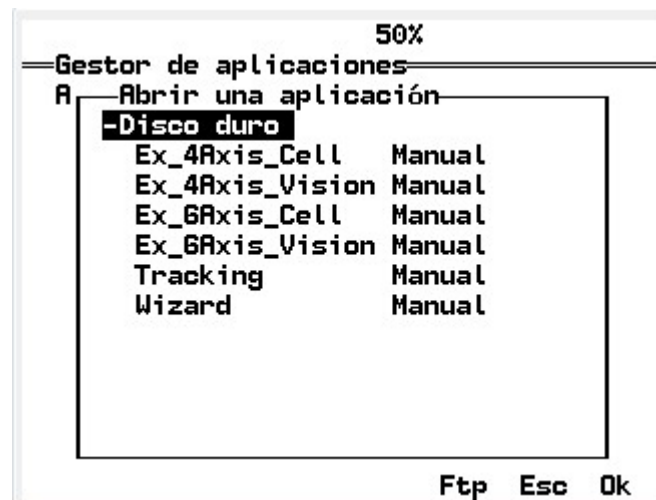
Funciones varias

- getConvSpeed(nCnvNb,l_nSpeed,l_nError)
Devuelve la velocidad de la cinta transportadora nCnvNb.
- getErrorMessage (&x_sMessage)
Esgta función devuelve u mensaje mas claro del ultimo código de error.
- kill ()
Para todas las tareas de seguimiento.
getObjectLatch (nCnvNb,nLatchedValue,l_nError)
Devuelve la posición de encoder del disparo (Latch) del objeto de la cola de la cinta transportadora nCnvNb.
- sendToNextRobot(sSocketClient, nTimerOut, nID, trObjectPos, NLatchedValue, l_nError)
Este programa envía un telegrama mediante cliente socket TCP/IP al siguiente robot con toda la información de la detección de la pieza.

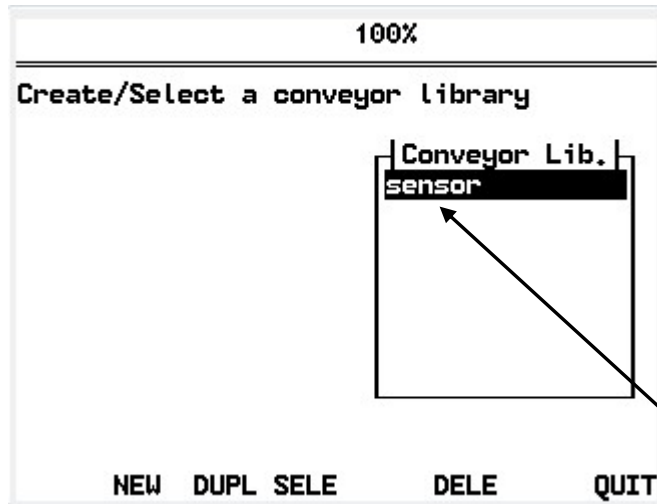
EJEMPLO / PLANTILLAS VALTRACK

El Valtrack lleva una serie de ejemplos / plantillas que nos servirán de base para nuestro programa:

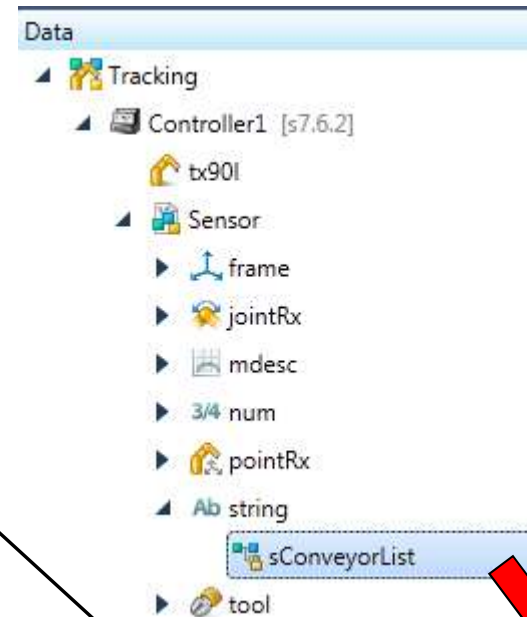
- Ex_4Axis_Cell: Scara con fotocélula.
- Ex_4Axis_Vision: Scara con cámara.
- Ex_6Axis_Cell: Antropomórfico con fotocélula.
- Ex_6Axis_Vision: Antropomórfico con cámara.



SELECCIÓN LIBRERÍA CONVEYOR

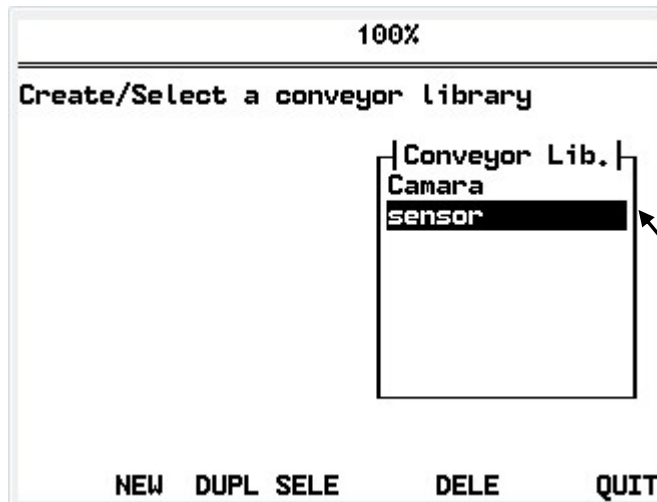


Wizard Valtrack

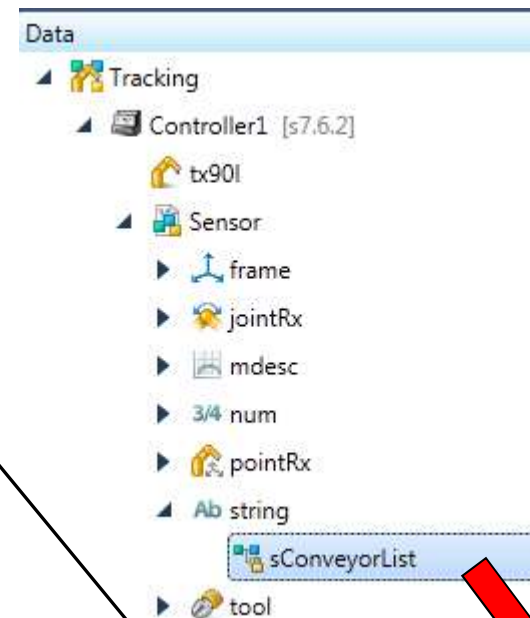


| Index | string |
|-------|--------|
| 0 | sensor |

Valtrack puede trabajar con más de una cinta transportadora.



Wizard Valtrack

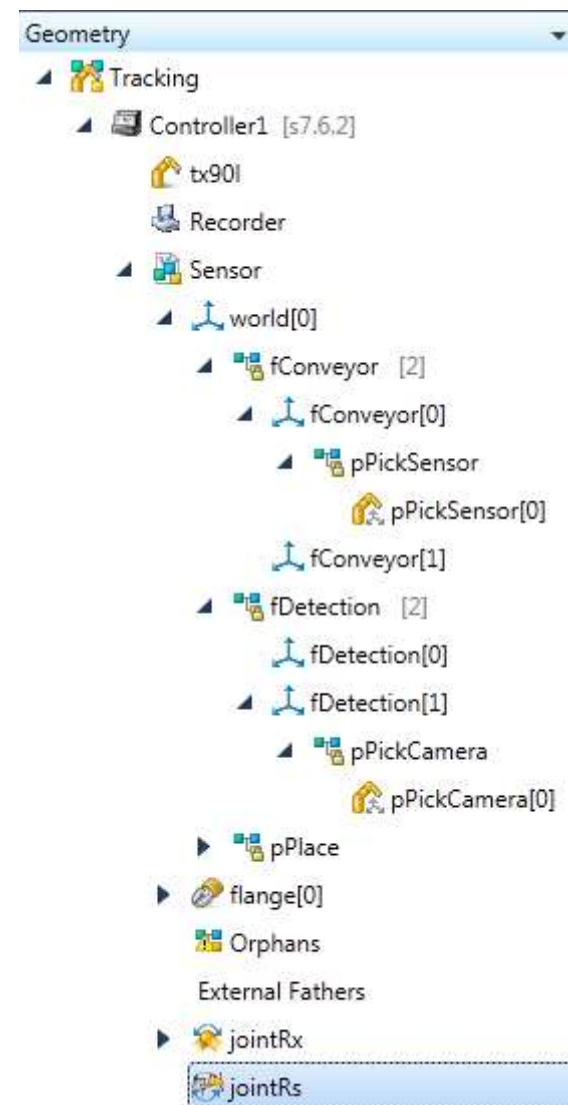



| Index | string |
|-------|--------|
| 0 | sensor |
| 1 | Camara |

En el caso de tener dos cintas, tendremos que añadir:

- Un elemento al plano cinta: fConveyor.
- Un elemento al plano detector: fDetection.
- Un elemento al sConveyorList.
- Un punto nuevo.

En el ejemplo adjunto, la librería 0 es con sensor (punto en el plano fConveyor) y la librería 1 es con cámara (punto en el plano fDetection).





Stäubli Española
Reina Elionor, 178-1º
08205 - Sabadell (Barcelona)

Telf. +34 93 720 54 08
Fax +34 93 712 42 56