

Aplicativo PSA Robots ABB

Índice

<u>1. Estructura del Programa.....</u>	1
<u>2. Tipos de rutinas:.....</u>	2
A. Rutina de trabajo tipo:	2
B. Rutinas auxiliares:	3
<u>3. Aplicaciones:.....</u>	5
<u>4. Particularidades de las aplicaciones más utilizadas:.....</u>	6
A. Manipulación (M)	6
B. Soldadura Neumática (SWN)	8
C. Soldadura Eléctrica (SWE).....	10
D. Soldadura Arco (MAG).....	10
E. Encolado (Ed/Ev).....	11
F. Pernos (G)	13
G. Cambio de herramienta:.....	14
<u>5.- Particularidades aplicativo:</u>	16
A. Anticolisiones:.....	16
B. Bifurcaciones:.....	17

1. Estructura del Programa.

Partiendo de la “Main”, entra en “R_PROGRAMA”, donde espera una orden y selecciona la rutina adecuada a la orden.

Ejemplo de “R_PROGRAMA” de un robot de manipulación:

```
PROC R_PROGRAMA ()
    ! PROGRAMA PRINCIPAL (Descripción rutina)
    !Lógica 26/01/05
    RPuntero\strRutina:="R_PROGRAMA"; (Memoriza la rutina en la ruta del puntero)
    VelSet 100,500;
    AccSet 100,100;
    WHILE DOutput(do100)=0 DO (Si do100=0, significa que el robot puede no estar en reposo, ya que al finalizar una rutina y llegar a reposo se pone esa señal a 0. Por si acaso, da fallo para que se verifique)
        RFalloAplicacion 2;
    ENDWHILE
    RValorColision TRUE,80; (Activa la supervisión de movimiento con el valor que le metamos)
    MoveJ p_reposo,v200,fine,t_garra; (Movimiento a posición de reposo (origen, home))
    R_INIT_SIGN; (Rutina de inicialización de señales)
    REsperaPrograma; (Rutina de espera orden del PLC)
    nPrograma:=gi_SELECCION_PRG;
    TEST nPrograma (Analizamos la orden y llamamos a la rutina solicitada. La rutina puede ser auxiliar o de trabajo)
    (Las primeras hasta el caso 71 son auxiliares)
```

```

CASE 24:
    RTextoTpu\strModelo:="MANTENIMIENTO GARRA
    1"\strTrayect:="R_MTO_GARRA1"; (Cargamos el nombre y la descripción de la rutina para
    mostrarla por pantalla)
    nVelocidad:=500; (Velocidad a la que queremos limitar el robot dentro de esta rutina)
    R_MTO_GARRA; (Llamada a la rutina correspondiente)
CASE 25:
    RTextoTpu\strModelo:="CONTROL DE POSICION GARRA
    1"\strTrayect:="R_C_POS_GARRA1";
    nVelocidad:=500;
    R_C_POS_GARRA;
CASE 71:
    RTextoTpu\strModelo:="3 PUERTAS"\strTrayect:="R_TRABAJO71";
    nVelocidad:=1000;
    R_TRABAJO71;
ENDTEST
RPuntero; (Actualiza la ruta del puntero)
RETURN;
ENDPROC

```

2. Tipos de rutinas:

Son 3:

- **De Trabajo:** Rutinas de soldadura, manipulación, etc....
- **Auxiliares:** De mantenimiento.
- **De Servicio:** Propias del robot.

A. Rutina de trabajo tipo:

```

PROC R_TRABAJOXX() (Nombre rutina de trabajo. Compuesto por "R_TRABAJO"+"XX" dos números. Empieza
en la 71)
!ROUTINA TRABAJO XXX (Descripción del trabajo)
!Lógica xx/xx/xx (Fecha programación)
RPuntero\strRutina:="R_TRABAJOXX"; (Metemos el nombre de la rutina, para memorizar la ruta)
RInicioTrabajo nVelocidad; (Variable de limitación de velocidad)
RInicioCiclo 0, "(Escribir acción a realizar)"; (Inicio de ciclo general)
(Entre el inicio de ciclo 0 general y el 1 se pueden insertar movimientos de aproximación a zona de trabajo)
RInicioCiclo 1, "(Escribir acción a realizar)"; (Espera inicio ciclo de trabajo)
Reset do_FUERA_TRAN1; (Ponemos a cero la señal para indicar que entramos en la zona de trabajo)
ClkReset clock1; (Reseteo del reloj de control de tiempo ciclo)
ClkStart clock1; (Inicio del reloj de control de tiempo ciclo)
MoveJ p_reposo, v3000, fine, t_Herramienta;
!Movimientos
(Movimientos y acciones necesarias en este primer ciclo)
PulseDO\PLength:=0.5, do_FIN_CICLO1; (Damos el fin de ciclo para que el PLC continúe el proceso)
Set do_FUERA_TRAN1; (Al acabar y salir de la zona de interferencia, ponemos la señal a 1)
RInicioCiclo 2, "(Escribir acción a realizar)"; (Inicio de un segundo ciclo si fuese necesario)
Reset do_FUERA_TRAN2; (Ponemos a cero la señal para indicar que entramos en la zona de trabajo)
!Movimientos
(Movimientos y acciones necesarias en este segundo ciclo)

```

```
PulseDO\PLength:=0.5,do_FIN_CICLO2; (Fin de ciclo de trabajo)
Set do_FUERA_TRAN2; (Al acabar y salir de la zona de interferencia, ponemos la señal a 1)
PulseDO\PLength:=0.5,do_FIN_CICLO0; (Fin de ciclo general)
ClkStop clock1; (Para reloj control de tiempo ciclo)
nReloj:=ClkRead(clock1); (Carga del tiempo ciclo en la variable correspondiente para mostrarlo por pantalla)
Set do100; (Señal que indica que el robot está en posición de reposo, se pone a 1, sólo cuando el robot está en la posición)
RPuntero; (Actualización ruta del puntero)
RETURN;
ENDPROC
```

Las señales que indican que estamos en una zona de trabajo o interferencia con los medios de la zona, son las salidas denominadas como "do_FUERA_TRAN1", donde el "1" se sustituye por el número de zona de trabajo. Como norma se pone la señal a "0" cuando entramos en la zona y a "1" cuando salimos de la zona. La señal de salida de zona hay que darla al mismo tiempo que el pulso de fin de ciclo de trabajo. En caso de tener que mover algo en la zona de trabajo mientras está el robot dentro se pone la señal de fin de ciclo de trabajo a "1" con un "Set", teniendo que permanecer el robot parado hasta que nos indiquen que el movimiento ha terminado, en ese momento ponemos a "0" con un "Reset" la señal de fin de ciclo y posteriormente podemos continuar con el trabajo.

Ejemplo: Soldamos en una maqueta y sin salir de ella, se mueve algún apriete, después continuamos soldando. Se haría de la siguiente manera:

```
...
RInicioCiclo 1,"Soldar";
Reset do_FUERA_TRAN1; (Ponemos a cero la señal para indicar que entramos en la zona de trabajo)
!Movimientos
(Movimientos de soldadura)
Set do_FIN_CICLO1; (Ponemos el fin de ciclo a "1" indicando que pueden moverse los aprietes)
RInicioCiclo 2,"Continuar soldando"; (Inicio de un segundo ciclo de soldadura. Esperamos la confirmación, que nos la darán cuando hay finalizado el movimiento)
Reset do_FIN_CICLO1; (Ponemos a cero la señal para indicar que continuamos trabajo e impedir que se pueda mover nada en la maqueta)
!Movimientos
(Movimientos y acciones necesarias en este segundo ciclo)
PulseDO\PLength:=0.5,do_FIN_CICLO2; (Fin de ciclo de trabajo)
Set do_FUERA_TRAN1; (Ponemos a "1" la señal para indicar que salimos de la zona de trabajo)
...
```

B. Rutinas auxiliares:

Listado general, no existen todas en todos los robots, sólo las propias del aplicativo del robot.

Port a robot	Denominación trayectoria auxiliar	Trayectoria robot
1	Rodado automático pinza 1	R_ROD_AUT_P1

2	Rodado manual pinza 1	R_ROD_MAN_P1
3	Mantenimiento pinza 1	R_MTO_P1
4	Refrigeración pinza 1	R_REFRIG_P1
5	Control de posición pinza 1	R_C_POS_P1
6	Control de soldadura pinza 1	R_CESUD_P1
7	Cambio de electrodos automático pinza 1	R_CAMBIO_AUT_CAB
8	Control alineado brazos pinza	R_CONTRL_ALIN
9	Reserva	
10	Reserva	
11	Rodado automático pinza 2	R_ROD_AUT_P2
12	Rodado manual pinza 2	R_ROD_MAN_P2
13	Mantenimiento pinza 2	R_MTO_P2
14	Refrigeración pinza 2	R_REFRIG_P2
15	Control posición pinza 1 y pinza 2	R_C_POS_P1P2
16	Control soldadura pinza 2	R_CESUD_P2
17	Cambio de electrodos automático pinza 2	R_CAMBIO_AUT_CAB 2
18	Rodado automático pinza 3	R_ROD_AUT_P3
19	Rodado manual pinza 3	R_ROD_MAN_P3
20	Refrigeración pinza 3	R_REFRIG_P3
21	Control soldadura pinza 3	R_CESUD_P3
22	Rodado manual pinza 1 y pinza 2	R_ROD_MAN_P1P2
23	Mantenimiento pinza 1 y pinza 2	R_MTO_P1P2
24	Mantenimiento garra 1* (para una herramienta)	R_MTO_GARRA1*
25	Control posición garra 1* (para una herramienta)	R_C_POS_GARRA1*
26	Mantenimiento garra 2* (para dos herramientas)	R_MTO_GARRA2*
27	Control posición garra 2* (para dos herramientas)	R_C_POS_GARRA2*
28	Purga automática pistolas de encolado	R_PURGA_AUT
29	Purga manual pistola 1 de encolado	R_PURGA_MAN_P1
30	Purga manual pistola 2 de encolado	R_PURGA_MAN_P2
31	Mantenimiento pistola robot encolado	R_MTO_ENCOLADO
32	Control de posición pistolas de encolado	R_C_POS_PISTOLAS
33	Reserva	
34	Reserva	
35	Mantenimiento pernos P1 y P2	R_MTO_PERNOS
36	Control de posición pistola pernos	R_C_POS_PERNOS
37	Control de soldadura pernos	R_CESUD_PERNOS
38	Reserva	
39	Reserva	
40	Control de posición antorcha arco	R_C_POS_PUNTERO
41	Control de soldadura arco	R_CESUD_ARCO
42	Mantenimiento antorcha arco	R_MTO_ARCO
43	Limpieza antorcha arco	R_LIMPIEZA
44	Cambio boquilla antorcha soldadura arco	R_CAMB_BOQUILLA
45	Limpieza y fresado antorcha arco	R_LIMPIA_C_FRESA
46	Reserva	
47	Control clinchado P1	R_CESUD_CLIN_P1

48	Mantenimiento pinza clinchado P1	R_MTO_CLINCHA
49	Control de posición pistola 1 clinchado	R_C_POS_CLINC_P1
50	Mantenimiento pinzas clinchado P1 y P2	R_MTO_CLINC_P1P2
51	Control de posición pistola 2 clinchado	R_C_POS_CLINC_P2
52	Dejar cambiador automático sin herramienta	RtrabajarConHerr 0
53	Reserva	
54	Reserva	
55	Mantenimiento cabezal marcado VIN	R_MTO_VIN
56	Control de posición cabezal marcado VIN	R_C_POS_VIN
57	Realiza comprobación del marcado del cabezal	R_T_EST_MARCADO
58	Reserva	
59	Mantenimiento torcha plasma	R_MTOPLASMA
60	Control de posición torcha plasma	R_C_POS_PLASMA
61	Sincronización de levas electrónicas en puestos con operario	
62	Reserva	
63	Reserva	

Ejemplo de rutina auxiliar, Rodado manual de pinza de soldadura:

```

PROC R_ROD_MAN_P1()
  !RODADO MANUAL PINZA 1
  !Lógica 02/10/07
  RPuntero\strRutina:="R_ROD_MAN_P1";
  RInicioPrgAux nVelocidad;
  RInicioCiclo 0,"Realizar Cambio de Electrodos P1";
  RInicioCiclo\Aux,1,"Realizar Cambio de Electrodos P1";
  RActivarInterrup 1;
  !Movimientos de Reposo a Rodado Manual
  MoveJ p_reposo,v500,fine,t_pinza1;
  MoveJ pRodManP1,v500,fine,t_pinza1;
  RAbrirPinza 1;
  RPulsaAuxiliares 1;
  RAbrirPinza 1\nPrgApertura:=301;
  RResetPosiRodMan;
  RRazContadores;
  !Movimientos a Reposo
  RTextoTpu\strAccion:"Robot a Reposo";
  MoveJ p_reposo,v500,fine,t_pinza1;
  Set current_enable;
  RFinPrgAux;
  RPuntero;
  RETURN;
ENDPROC

```

3. Aplicaciones:

- Soldadura Arco (MAG)
- Cambio de Herramienta
- Clinchado
- Corte Plasma
- Encolado

- Engatillado
- Manipulación (MH)
- Pernos
- SERV
- Soldadura Eléctrica (SWE)
- Soldadura Neumática (SWN)
- Marcado VIN

4. Particularidades de las aplicaciones más utilizadas:

A. Manipulación (M)

Para el tratamiento de la garra de manipulación desde el programa se divide en “Garras” que corresponden con los “distribuidores neumáticos” o los “equipos de vacío” instalados en dicha garra. Cada distribuidor puede actuar sobre varios aprietes los cuales van a actuar al mismo tiempo, pero el control sobre los detectores de apertura y cierre es individual.

Por norma a la hora de definir las salidas se ponen primero los equipos de vacío y después los distribuidores neumáticos.

A la hora de definir las entradas se ponen primero los detectores de presencia pieza, después las entradas del equipo de vacío y por último los detectores de apertura y cierre de los aprietes.

Ejemplo: Configuración de garra:

GARRA= (1 distribuidor con 2 aprietes) + (1 equipo de vacío) + (2 detectores de presencia pieza)
Garra 1 de aprietes Garra 2 de ventosas

Salidas:

Número de salida	Nombre salida
1	do_SUCCION2
2	do_SOPLADO2
3	do_CERRAR_GARRA1
4	do_ABRIR_GARRA1

Entradas:

Número de entrada	Nombre entrada
1	di_PRESEN_PIEZA1
2	di_PRESEN_PIEZA2
3	di_VACIO_GARRA2
4	Reserva
5	di_A1CERR_GARRA1
6	di_A1ABIE_GARRA1
7	di_A2CERR_GARRA1
8	di_A2ABIE_GARRA1

Rutina tipo de manipulación:

```

PROC R_TRABAJO71()
  !R_MANIPULACION
  !Lógica      05/07/05
  RPuntero\strRutina:="R_TRABAJO71";
  RInicioTrabajo nVelocidad;
  RInicioCiclo 0," ";
  RInicioCiclo 1,"Coger Pieza"; (Ciclo de cogida de pieza)
  Reset do_FUERA_TRAN1;
  ClkReset clock1;
  ClkStart clock1;
  MoveJ p_reposo,v3000,fine,t_garra;
  RAbrirGarra 1; (Rutina utilizada para abrir la garra que se le indica. Recordar que en caso de varios distribuidores puede ser necesario repetir la llamada a la rutina para el resto de garras)
(Se manda abrir las garras por si quedaron abiertas en una rutina anterior o por si se abrieron en manual, evitando así una colisión)
  !Movimientos
(Nos desplazamos a la posición de cogida)
  RCerrarGarra 1; (Rutina utilizada para cerrar la garra que se le indica. Recordar que en caso de varios distribuidores puede ser necesario repetir la llamada a la rutina para el resto de garras)
  GripLoad l_pieza; (Actualizamos los datos de carga del robot)
  RPresenciaPieza 1; (Rutina que comprueba que el detector de presencia pieza 1 está detectando, también hace un seguimiento de la detección. En caso de varios detectores es necesario llamar a la rutina con los números correspondientes)
  !Movimientos
(Movimientos fuera de la maqueta)
  Set do_FUERA_TRAN1;
  PulseDO\PLength:=0.5,do_FIN_CICLO1;
  RInicioCiclo 2,"Dejar Pieza"; (Ciclo de dejada de pieza)
  Reset do_FUERA_TRAN2;
  !Movimientos
(Movimientos a zona de dejada)
  RSinCtrlPrePieza 1; (Desactivamos el control de pieza)
  RAbrirGarra 1; (Abrimos garra)
  GripLoad load0; (Actualizamos los datos de carga)
  !Movimientos
(Pequeño movimiento separándonos de la pieza)
  RSinPieza 1; (Rutina que comprueba la ausencia de pieza en la garra)
  !Movimientos
(Movimientos de salida a reposo)
  MoveJ p_reposo,v3000,fine,t_garra;
  PulseDO\PLength:=0.5,do_FIN_CICLO2;
  PulseDO\PLength:=0.5,do_FIN_CICLO0;
  Set do_FUERA_TRAN2;
  ClkStop clock1;
  nRelej:=ClkRead(clock1);
  Set do100;
  RPuntero;
  RETURN;
ENDPROC

```

Un caso especial de la manipulación es el desapilado, que se utiliza para coger las piezas que vienen apiladas en cestones. Rutina tipo:

```

PROC R_DESAPILAR_P1P1()
  ! DESAPILAR PILA 1 PALET 1
  !Logica      17/03/06

```

```

RPuntero\strRutina:="R_DESAPILAR_P1P1";
RTextoTpu\strAccion:="Coger Pieza Pila 1 Palet 1";
!Movimiento Robot a encima Pila 1 Palet 1
MoveJ pEncimaPila1P1,v3000,z10,t_garra\WObj:=wobj0; (Movimiento encima del palet, es la primera posición que tenemos que grabar)
!Movimiento Robot a posicion cogida anterior si el Plc lo autoriza
IF di_INICIO_PALET=0 MoveL pAnteriorPieP1P1,v1000,z10,t_garra\WObj:=wobj0; (Si no es inicio de palet nos acercamos a la última posición de cogida para ganar tiempo)
!Movimiento de busqueda de la pieza
SearchL\PStop,di_PRESEN_PIEZA1,pCogidaPila1P1,pFondoPila1P1,v100,t_garra\WObj:=wobj0; (Búsqueda de la pieza y memorización de la posición. El punto "pFondoPila1P1" es el otro que tenemos que memorizar, ya que la búsqueda la hará en lineal desde el punto de cogida de la anterior pieza o si es cestón nuevo desde el punto encima del palet)
RCerrarGarra 1;
RPresenciaPieza 1;
!Memorizar punto inicio siguiente busqueda
pAnteriorPieP1P1:=Offs(pCogidaPila1P1,0,0,0); (Memoriza el último punto de cogida para aproximarse para la siguiente pieza)
!Movimientos de salida con la pieza
MoveL Offs(pCogidaPila1P1,0,0,100),v200,z10,t_garra\WObj:=wobj0; (Movimiento de salida)
!Movimiento Robot a encima Palet
MoveL pEncimaPila1P1,v500,z10,t_garra\WObj:=wobj0; (Punto encima del palet, el mismo del principio de la rutina)
bDesapiladoOk:=TRUE;
RTextoTpu\strAccion:=" ";
RPuntero;
RETURN;
ERROR (Gestor de errores que también hay que programar)
!Movimientos Robot a encima Palet
MoveL pEncimaPila1P1,v100,fine,t_garra\WObj:=wobj0;
IF ERRNO=ERR_WHLSEARCH THEN
    !Fallo Busqueda. Pieza no encontrada
    RFalloManipulaci 26; (Pieza no encontrada, podemos optar por repetir o saltar)
    IF nFkAplic=1 THEN
        !Opcion saltar
        !Movimientos Robot a reposo (Se programan los movimientos del punto encima palet a reposo)
        MoveJ p_reposo,v200,fine,t_garra;
        RSaltarDesapilad;
    ENDIF
    !Opcion repetir (Con esta opción reintentará la búsqueda)
    RETURN;
ELSEIF ERRNO=ERR_SIGSUPSEARCH THEN
    !Fallo Busqueda. Detector busqueda activado
    RFalloManipulaci 27; (Detector activado, podemos optar por repetir o saltar)
    IF nFkAplic=1 THEN
        !Opcion saltar
        !Movimientos Robot a reposo (Se programan los movimientos del punto encima palet a reposo)
        MoveJ p_reposo,v200,fine,t_garra;
        RSaltarDesapilad;
    ENDIF
    !Opcion repetir (Con esta opción reintentará la búsqueda)
    RETRY;
ENDIF
ENDPROC

```

B. Soldadura Neumática (SWN)

Soldadura por puntos con pinza neumática.

Las instrucciones de soldadura por puntos en Psa-Vigo son “SpotLi” y “SpotJi”, estructuradas de la siguiente manera:

SpotLi j10000, v200, B10000, gun1, t_pinza1 \WObj:=wobject;

0 1 2 3 4 5 6

- 0.- “SpotLi” con movimiento de aproximación lineal. La estructura es común con “SpotJi”
- 1.- Punto de soldadura. Los puntos se nombrarán con el número proporcionado por Psa precedido de una “j”. Ejemplo: punto 10000, se llamará “j10000”.
- 2.- Velocidad de aproximación al punto.
- 3.- Programa de soldadura. Los programas de soldadura correspondientes a cada punto se nombran con el número de punto precedido por una “B”. Ejemplo: al punto “j10000” le corresponde el programa de soldadura “B10000”.
- 4.- Número de pinza. Se indica por si hubiera más de una.
- 5.- Herramienta. Se nombrará como “t_pinza” + “(el número de pinza)”
- 6.- Objeto de trabajo (Opcional). Se utilizará el objeto de trabajo de la maqueta o puesto sobre el que se suelde, en caso de ser una pinza estacionaria se utiliza el “WobjEstacionario”.

Otros argumentos opcionales propios de la instrucción son:

- \Detect : Control Pegado electrodos si el control esfuerzo está anulado.
- \Compen2: Control de Compensación 2 de la Pinza. Para trabajar con la pinza al revés se activa una compensación neumática

Rutina tipo de soldadura:

```
PROC R_TRABAJO71 ()
  !TRAYECTORIA TRABAJO SWN
  !Lógica      25/05/04
  RPuntero\strRutina:="R_TRABAJO71";
  RInicioTrabajo nVelocidad;
  RInicioCiclo 0," ";
  RInicioCiclo 1,"Soldar Pieza";
  Reset do_FUERA_TRAN1;
  ClkReset clock1;
  ClkStart clock1;
  MoveJ p_reposo,v3000,fine,t_pinza1;
  RAbrirPinza 1; (Rutina de abrir la pinza que se indica. Por si es necesario para entrar en la pieza)
  !MOVIMIENTOS
  (Movimientos de aproximación a la zona de soldadura)
  RAproximarPinza 1; (Rutina de aproximar la pinza que se le indique)
  SpotLi j10000, v200, B10000, gun1, t_pinza1\WObj:=WobjOp1; (Soldadura)
  (Movimientos de salida de pieza hasta reposo)
  MoveJ p_reposo,v3000,fine,t_pinza1;
  RAbrirPinza 1;
  PulseDO\PLength:=0.5,do_FIN_CICLO1;
  Set do_FUERA_TRAN1;
  PulseDO\PLength:=0.5,do_FIN_CICLO0;
  ClkStop clock1;
  nReloj:=ClkRead(clock1);
  Set do100;
```

```

RPuntero;
RETURN;
ENDPROC

```

C. Soldadura Eléctrica (SWE)

Soldadura por puntos con pinza eléctrica. Básicamente igual a la neumática con la diferencia de la pinza.

Las instrucciones de soldadura por puntos en Psa-Vigo son “SpotLi” y “SpotJi”, estructuradas de la siguiente manera:

```

SpotLi j10000, v200, gun1, B10000, t_pinza1 \WObj:=wobject;
0         1         2         3         4         5         6

```

Los argumentos nº de pinza (3) y programa (4) van intercambiados con respecto a la soldadura neumática

- 0.- “SpotLi” con movimiento de aproximación lineal. La estructura es común con “SpotJi”
- 1.- Punto de soldadura. Los puntos se nombrarán con el número proporcionado por Psa precedido de una “j”. Ejemplo: punto 10000, se llamará “j10000”.
- 2.- Velocidad de aproximación al punto.
- 3.- Número de pinza. Se indica por si hubiera más de una.
- 4.- Programa de soldadura. Los programas de soldadura correspondientes a cada punto se nombran con el número de punto precedido por una “B”. Ejemplo: al punto “j10000” le corresponde el programa de soldadura “B10000”.
- 5.- Herramienta. Se nombrará como “t_pinza” + “(el número de pinza)”
- 6.- Objeto de trabajo (Opcional). Se utilizará el objeto de trabajo de la maqueta o puesto sobre el que se suelde, en caso de ser una pinza estacionaria se utiliza el “WobjEstacionario”.

Otro argumento opcional propio de la instrucción es:

\Compen2: Control de Compensación 2 de la Pinza. Para trabajar con la pinza al revés se activa una compensación neumática.

D. Soldadura Arco (MAG)

Instrucción propia de PSA-Vigo:

```

ArcLi\On, j10000i, v3000, sm0, wd0, wv0, t_Antorcha \WObj:=Wobj0;
0         1         2         3         4         5         6         7         8

```

0.- Movimiento lineal

- 1.- Argumento opcional con varias posibilidades:
 - Sin argumento: Continúa la soldadura iniciada anteriormente.
 - \On: empieza la soldadura al llegar al punto.
 - \Off: Termina la soldadura al llegar al punto.
 - \Bot: Suelta un punto al llegar al robtarget.
- 2.- Punto del cordón de soldadura. Los puntos se nombrarán con el número de cordón proporcionado por Psa precedido de una “j” y seguido de una “i” si es el punto inicial o de una “f” si es el final.

Ejemplo: cordón 10000, se llamarán:

Punto inicial: "j10000i".

Punto medio: "j10000_XX"

Punto final: "j10000f"

3.- Velocidad de aproximación al punto.

4.- "seamdata" de la soldadura.

5.- "welddata" de la soldadura.

6.- "weavedata" de la soldadura.

7.- Herramienta. Se nombrará como "t_Antorcha".

8.- Objeto de trabajo (Opcional). Se utilizará el objeto de trabajo de la maqueta o puesto sobre el que se suelde, en caso de ser una antorcha estacionaria se utiliza el "WobjEstacionario".

Rutina tipo de soldadura Arco:

```
PROC R_TRABAJO71 ()
  !RUTINA TRABAJO SOLDADURA ARCO
  !Lógica 27/08/04
  RPuntero\strRutina:="R_TRABAJO71";
  RInicioTrabajo nVelocidad;
  RInicioCiclo 0, " ";
  RInicioCiclo 1, "Soldar Pieza";
  Reset do_FUERA_TRAN1;
  RPantIniSIO;
  ClkReset clock1;
  ClkStart clock1;
  MoveJ p_reposo, v3000, fine, t_Antorcha;
  !Movimientos
  (Movimientos a zona de soldadura)
  ArcLi\On, j10000i, v3000, sm0, wd0, wv0, t_Antorcha\WObj:=Wobj0; (Inicio cordón )
  . . . .
  ArcLi\Off, j10000f, v3000, sm0, wd0, wv0, t_Antorcha\WObj:=Wobj0; (Final cordón )
  (Movimientos de zona de soldadura a reposo)
  MoveJ p_reposo, v3000, fine, t_Antorcha;
  PulseDO\PLength:=0.5, do_FIN_CICLO1;
  Set do_FUERA_TRAN1;
  PulseDO\PLength:=0.5, do_FIN_CICLO0;
  ClkStop clock1;
  nReloj:=ClkRead(clock1);
  Set dol00;
  RPuntero;
  RETURN;
ENDPROC
```

E. Encolado (Ed/Ev)

Aplicación de colas o mástico en piezas.

La instrucción propia de PSA-Vigo para esta aplicación es "EncoL", cuya estructura es la siguiente:

EncoL 1\nDOn:=10, j10000, v300, z50, tPistolaEncolad1, wobj0;
 0 1 2 3 4 5 6 7 8

0.- Instrucción de aplicación de cola con movimiento de aproximación lineal.

1.- Número de pistola de cola.

2.- Argumento opcional para aplicar o no aplicar cola, puede ser:

\nDOn: Comenzar a aplicar.

\nDOff: Parar de aplicar.

- 3.- Distancia de anticipación de la señal tanto para argumento “nDOn”, como “nDOff”.
- 4.- Punto del cordón de cola2. Los puntos se nombrarán con el número de cordón proporcionado por Psa precedido de una “j” y seguido de una “i” si es el punto inicial o de una “f” si es el final, a los intermedios se les añadirá “_x” siendo “x” números consecutivos. Ejemplo: cordón 10000, se llamarán:
Punto inicial: “j10000i”.
Puntos medios: “j10000_1”, “j10000_2”...
Punto final: “j10000f”.
- 5.- Velocidad del movimiento.
- 6.- Zona.
- 7.- Herramienta, se denominará como “tPistolaEncolad” + “(número de pistola)”.
- 8.- Objeto de trabajo (Obligatorio). Se utilizará el objeto de trabajo de la maqueta o puesto sobre el que se trabaje, en caso de ser una pistola estacionaria se utiliza el “WobjEstacionario”.

Otros argumentos opcionales propios de la instrucción son:

- \ J: Movimiento no lineal.
- \num nPrgCaudal: Programa de caudal a cargar en el equipo de cola, distinto de “1” que es el programa por defecto.
- \num nDCaudal: Distancia a anticipar la carga del programa del caudal que se indique en el argumento “nPrgCaudal”, o si no existe, del programa por defecto “1”.

```
PROC R_TRABAJO65()
  !R_ENCOLADO 1 PISTOLA
  !Logica 25/11/05
  RPuntero\strRutina:="R_TRABAJO65";
  RInicioTrabajo nVelocidad;
  RInicioCiclo 0, " ";
  RInicioCiclo 1, "Coger Pieza";
  Reset do_FUERA_TRAN1;
  ClkReset clock1;
  ClkStart clock1;
  MoveJ p_reposo, v3000, fine, tPistolaEncolad1;
  !MASTICO Y COLA
  VelSet 100, 250; (Limitamos la velocidad de aproximación a la aplicación de cola)
  !Movimientos
  RCodigoPieza 1, 1; (Rutina de inicialización en la que se le indica el número de pistola y el número de pieza.)
  ! Pistola 1
  ! Mastico
  ! Punto ANTERIOR al bucle de encolado
  MoveJ p_anterior, v500, z20, tPistolaEncolad1\WObj:=wobj0; (Punto anterior al encolado, se usa de enlace si hay que repetir la pieza)
  bRepetirPieza:=TRUE;
  WHILE bRepetirPieza=TRUE DO
    RDosificadorOk 1; (Rutina que comprueba que el dosificador está OK)
    VelSet 100, 100; (Limitamos la velocidad durante la aplicación de cola)
    ! Cordon 1 Mastico (Cordón 1)
    EncoL 1\nDOn:=0, j10000_i, v300, z50, tPistolaEncolad1, wobj0;
    EncoL 1\nDOff:=0, j10000_f, v300, z50, tPistolaEncolad1, wobj0;
    MoveJ p10, v500, z20, tPistolaEncolad1\WObj:=wobj0;
    ! Cordon 2 Mastico (Cordón 2)
    EncoL 1\nDOn:=10, j20000_i, v300, z50, tPistolaEncolad1, wobj0;
    EncoL 1\nPrgCaudal:=2, j20000_1, v300, z50, tPistolaEncolad1, wobj0;
    EncoL 1\nPrgCaudal:=0, j20000_2, v300, z50, tPistolaEncolad1, wobj0;
    EncoL 1\nDOff:=0, j20000_f, v300, z50, tPistolaEncolad1, wobj0;
```

```

MoveL p20,v500,z5,tPistolaEncolad1\WObj:=wobj0;
RFinEncolado 1; (Indica el final de la aplicación)
!Comprobacion Aplicacion Producto
IF bFalloEncolado=TRUE THEN (Comprobación de cordon bueno)
    !Aplicacion Producto Con Fallo
    ! Movimientos a Comprobacion Encolado (Si es malo se lleva a una posición para mostrarlo)
    MoveJ pFalloPistola1_2,v200,z50,tPistolaEncolad1\WObj:=wobj0;
    MoveJ pFalloPistola1_1,v200,z50,tPistolaEncolad1\WObj:=wobj0;
    MoveJ pFalloPistola1,v200,fine,tPistolaEncolad1\WObj:=wobj0;
    RMostrarFallo 1; (Espera confirmación externa para continuar, le pueden indicar que repita la pieza o que la salte ignorando el fallo)
    IF bRepetirPieza=TRUE THEN (Si hay que repetir pieza)
        RCodigoPieza 1,1; (Cargamos e código de pieza de nuevo)
        ! REPETIR ENCOLADO
        ! Movimientos a punto ANTERIOR al bucle de encolado (Se enlaza con el punto anterior al encolado)
    ELSE
        ! SALTAR ENCOLADO (si hay que saltar la pieza)
        ! Movimientos a punto POSTERIOR al bucle de encolado ( se enlaza con el punto posterior al encolado)
        MoveJ pFalloPistola1_1,v200,z50,tPistolaEncolad1;
        MoveJ pFalloPistola1_2,v200,z50,tPistolaEncolad1;
    ENDIF
ELSE
    !Aplicacion Producto Sin Fallo
    bRepetirPieza:=FALSE;
ENDIF
ENDWHILE
! Punto POSTERIOR al bucle de encolado
MoveJ p_posterior, v500,z20,tPistolaEncolad1\WObj:=wobj0; (Punto posterior)
!Petición Comprobacion cordones de Encolado
IF di_MUESTRA_PIEZA=1 THEN (Si hay petición de muestra de cola)
    !Movimientos a punto de Muestra
    MoveJ p_Muestra_2,v200,z200,tPistolaEncolad1\WObj:=wobj0;
    MoveJ p_Muestra_1,v200,z200,tPistolaEncolad1\WObj:=wobj0;
    MoveJ p_Muestra,v200,fine,tPistolaEncolad1\WObj:=wobj0; (Punto de muestra, ha de ser un lugar visible desde el exterior de la instalación)
    RMostrarPieza; (Espera confirmación externa para continuar)
    !Movimientos del punto de muestra al punto anterior
    MoveJ p_Muestra_1,v200,z200,tPistolaEncolad1\WObj:=wobj0;
    MoveJ p_Muestra_2,v200,z200,tPistolaEncolad1\WObj:=wobj0;
ENDIF
!Movimientos
MoveJ p_reposo,v3000,fine,tPistolaEncolad1\WObj:=wobj0;
PulseDO\PLength:=0.5,do_FIN_CICLO1;
PulseDO\PLength:=0.5,do_FIN_CICLO0;
Set do_FUERA_TRAN2;
ClkStop clock1;
nReloj:=ClkRead(clock1);
Set dol00;
RPuntero;
RETURN;
ENDPROC

```

F. Pernos (G)

Soldadura de pernos o Tucker controlada desde el robot.

La instrucción propia de PSA-Vigo para esta aplicación es “SoldPernoL”, cuya estructura es la siguiente:

SoldPernoL 1, j1000, v3000, t_PistolaPernos \WObj:=Wobj0;

1 2 3 4 5 6

- 1.- Instrucción de soldadura de pernos con aproximación lineal.
- 2.- Número de pistola o cabezal de pernos.
- 3.- Punto de soldadura del perno. Los puntos se nombrarán con el número proporcionado por Psa precedido de una “j”. Ejemplo: punto 10000, se llamará “j10000”.
- 4.- Velocidad del movimiento.
- 5.- Herramienta, se denominará como “t_PistolaPernos” + “(número de pistola)”.
- 6.- Objeto de trabajo (Opcional). Se utilizará el objeto de trabajo de la maqueta o puesto sobre el que se suelde, en caso de ser una pistola estacionaria se utiliza el “WobjEstacionario”.

Otro argumento opcional propio de la instrucción es:

\num nSigPerno: Precarga del siguiente perno a soldar.

Rutina tipo de soldadura de pernos:

```
PROC R_TRABAJO71 ()
    !RUTINA TRABAJO SOLDADURA PERNOS
    !Lógica      21/02/05
    RPuntero\strRutina:="R_TRABAJO71";
    RInicioTrabajo nVelocidad;
    RInicioCiclo 0, " ";
    RInicioCiclo 1, "Soldar Pieza";
    Reset do_FUERA_TRAN1;
    ClkReset clock1;
    ClkStart clock1;
    MoveJ p_reposo, v3000, fine, t_PistolaPernos;
    RActivarCtrlPern; (Rutina de activación del control del proceso de soldadura de pernos)
    RCargaPerno 1; (Rutina de carga de perno en el cabezal indicado)
    !Movimientos
    (Movimientos a zona de soldadura)
    SoldPernoL 1, j1000, v3000, t_PistolaPernos; (Instrucción de soldadura)
    (Movimientos a reposo)
    MoveJ p_reposo, v3000, fine, t_PistolaPernos;
    PulseDO\PLength:=0.5, do_FIN_CICLO1;
    Set do_FUERA_TRAN2;
    PulseDO\PLength:=0.5, do_FIN_CICLO0;
    ClkStop clock1;
    nReloj:=ClkRead(clock1);
    Set do100;
    RPuntero;
    RETURN;
ENDPROC
```

G. Cambio de herramienta:

Cuando el robot necesita cambiar la herramienta o simplemente dejarla.

Rutina tipo:

```
PROC R_TRABAJO71()
  !MANIPULACION CON GARRA 1 DE CAMBIO AUTOMATICO
  !Lógica      13/09/05
  RPuntero\strRutina:="R_TRABAJO71";
  RInicioTrabajo nVelocidad;
  RInicioCiclo 0,"Coger Herramienta";
  !Movimientos de cogida herramienta 1 si no
  !esta cogida
  RTrabajarConHerr (número de herramienta con la que queremos trabajar);
  (Rutina que coge la herramienta que le indiquemos si no está cogida. Si tuviera otra cogida, primero deja la que tiene y después coge la que le indicamos)
  RInicioCiclo 1,"Trabajo";
  Reset do_FUERA_TRAN1;
  ClkReset clock1;
  ClkStart clock1;
  MoveAbsJ p_reposo,v3000,fine,t_Herramiental; (Hay que tener cuidado con la herramienta utilizada para el reposo. Normalmente el punto de reposo será un joinTarget)
  (Realizamos el trabajo que queramos con la herramienta seleccionada)
  !Movimientos de dejada herramienta
  RTrabajarConHerr 0; (Al utilizar la herramienta 0, significa que deje la herramienta que tenga cogida)
  MoveAbsJ p_reposo,v3000,fine, t_CambioHerram; (Al dejar la herramienta utilizamos un tool con las masas bien definidas)
  PulseDO\PLength:=0.5,do_FIN_CICLO1;
  PulseDO\PLength:=0.5,do_FIN_CICLO0;
  Set do_FUERA_TRAN1;
  ClkStop clock1;
  nReloj:=ClkRead(clock1);
  Set do100;
  RPuntero;
  RETURN;
ENDPROC
```

Las rutinas a programar son las de cogida y dejada de todas las herramientas que tengamos disponibles. La rutina tipo de cogida herramienta es:

```
PROC R_COGER_HERRAMI1()
  ! COGER HERRAMIENTA 1 (Pinza, garra, ...)
  !Lógica      01/09/05
  RPuntero\strRutina:="R_COGER_HERRAMI1";
  RTextoTpu\strAccion:="Coger Herramienta 1";
  RRobotSinHerrami; (Se comprueba que el robot está sin herramienta y puede coger la que le indicamos)
  RAbrirTapaHerram 1; (Abre la tapa de protección del soporte de la herramienta)
  !Movimientos a posición cogida herramienta
  MoveAbsJ p_reposo,v200,fine,t_CambioHerram; (Movimientos con tool adecuado)
  MoveJ p_Herram1_1,v200,z200,t_CambioHerram; (Los puntos de aproximación se denominan como "p_HerramX_Y", donde "X" es el número de herramienta e "Y" el número de punto)
  MoveJ p_Herram1_2,v200,z1,t_CambioHerram;
  MoveL p_Coger_Herram1,v100,fine,t_CambioHerram; (El punto de cogida se denomina "p_Coger_HerramX", donde "X" es el número de herramienta)
  RCerrarCambioHer; (Cierre del cambio de herramienta)
  RActivaModuDNET 1; (Activa los módulos devicenet que haya en la herramienta)
  ! Activar Tpc de la herramienta (Pinza, garra, ...)
  ! en instrucciones movimiento siguientes
  MoveL p_Herram1_3,v100,fine,t_Herramiental; (Cambiamos al tool de la herramienta que cogimos)
```

```

RRobotConHerrami 1; (Se comprueba que está la herramienta cogida)
MoveL p_Herrai_4,v200,z10,t_Herramienta1;
RTextoTpu\strAccion:=" ";
RPuntero;
RETURN;
ENDPROC

```

La rutina tipo de dejada herramienta es:

```

PROC R_DEJAR_HERRAMI1()
! DEJAR HERRAMIENTA 1 (Pinza, garra, ...)
!Lógica 09/11/05
RPuntero\strRutina:="R_DEJAR_HERRAMI1";
! Activar Tcp de la herramienta (Pinza, garra, ...)
! en instrucciones movimiento siguientes
!Movimientos a posición dejada herramienta
RTextoTpu\strAccion:="Dejar Herramienta 1";
MoveJ p_Herrai_4,v200,z10,t_Herramienta1;
MoveL p_Herrai_3,v100,fine,t_Herramienta1;
MoveL p_Dejar_Herrai,v100,fine,t_Herramienta1; (El punto de dejada se denomina
"p_Dejar_HerraX", donde "X" es el número de herramienta)
RDesactivaMDNET 1; (Desactiva los módulos devicenet que haya en la herramienta)
RAbrirCambioHerr; (Abre el cambio de herramienta)
MoveL p_Herrai_2,v100,fine,t_CambioHerram; (Cambiamos al tool adecuado)
RRobotSinHerrami; (Se comprueba que el robot está sin herramienta)
MoveJ p_Herrai_1,v200,z200,t_CambioHerram;
RTextoTpu\strAccion:="Robot a Reposo";
MoveAbsJ p_reposo,v200,fine,t_CambioHerram;
RCerrarTapaHerra 1; (Cierra la tapa de protección del soporte de la herramienta)
RPuntero;
RETURN;
ENDPROC

```

5.- Particularidades aplicativo:

A. Anticolisiones:

Son las zonas de interferencia entre robots, o entre un robot y algún elemento móvil, como una mesa giratoria o un transportador.

La gestión la lleva el PLC, desde el robot nos limitamos a pedir autorización de entrada, informar que estamos dentro e indicar cuando salimos. La instrucción de entrada en zona de colisión es la siguiente:

```

REntZonaColision 1, p1, v200, z1, t_garra, WObj:=wobject;
0 1 2 3 4 5 6

```

- 0.- "REntZonaColision" con movimiento de aproximación por ejes.
- 1.- Número de anticolisión, a ponerse de acuerdo con el automatista.
- 2.- Punto en el cual se realiza la petición de entrada, ha de estar fuera de la zona de interferencia.
- 3.- Velocidad de aproximación al punto.

- 4.- Zona para el punto de petición.
- 5.- Herramienta.
- 6.- Objeto de trabajo. Es obligatorio con lo que si no se quiere definir ninguno, habrá que poner "wobj0".

Otro argumento opcional propio de la instrucción es:

\ L: Aproximación al punto de petición con movimiento lineal.

La instrucción de salida de zona de colisión es la siguiente:

```
RSalZonaColision 1, p1, v200, fine, t_garra, WObj:=wobject;
                  0      1   2      3      4      5      6
```

- 0.- "RSalZonaColision" con movimiento de aproximación por ejes.
- 1.- Número de anticolidión, a ponerse de acuerdo con el automatista.
- 2.- Punto en el cual se informa de la salida de la zona de interferencia, ha de estar fuera de la zona.
- 3.- Velocidad de aproximación al punto.
- 4.- Zona para el punto de salida. Como norma será un "fine" o se garantizará perfectamente que no pueda haber problemas de interferencia
- 5.- Herramienta.
- 6.- Objeto de trabajo. Es obligatorio con lo que si no se quiere definir ninguno, habrá que poner "wobj0".

Otro argumento opcional propio de la instrucción es:

\ L: Aproximación al punto de salida con movimiento lineal.

B. Bifurcaciones:

Al llegar a una espera de inicio ciclo, se puede dar más de una opción, concretamente hasta cuatro. La instrucción de espera de señal de inicio ciclo "RInicioCiclo" y la forma de tratar el caso es la siguiente:

```
RInicioCiclo 1\nClB2:=3\nClB3:=5\nClB4:=7, "Soldar o Coger o Transportador o
Sondeo"; (Según la señal de inicio ciclo que reciba del PLC, realizará una acción u otra)
TEST nBifurcacion (Variable a consultar para ver que trabajo debemos realizar, esto significa que la rutina
"RInicioCiclo" carga en "nBifurcación" un número del 1 al 4 según el inicio ciclo que se reciba)
CASE 1: (Si recibe el inicio ciclo que está puesto en primer lugar, realiza esta acción)
        !Soldar
CASE 2: (Si recibe el inicio ciclo que está puesto en segundo lugar, realiza esta acción)
        !Coger
CASE 3: (Si recibe el inicio ciclo que está puesto en tercer lugar, realiza esta acción)
        !Transportador
CASE 4: (Si recibe el inicio ciclo que está puesto en cuarto lugar, realiza esta acción)
        !Sondeo

ENDTEST
```