CAPÍTULO 2: Elaboración de la información

2.1 INTRODUCCIÓN.

En este capítulo se detallará la forma en la que se han extraído los datos mostrados por la interfaz gráfica, así como la estructura de datos con la que interacciona (la ficha de puestos).

Recordamos que estos datos consistían en qué palabras accede cada uno de los puestos de Inductel controlados por los programas autómatas; distinguiendo además en si las palabras son accedidas en lectura o escritura.

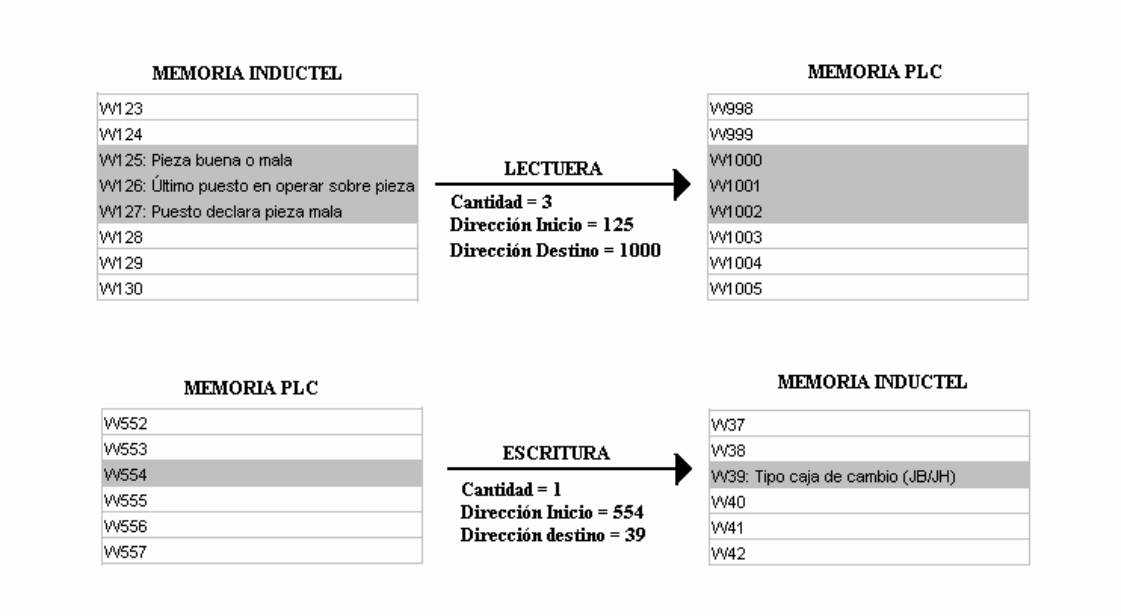


Figura 2.1: Acceso en lectura y escritura mediante Inductel.

En estas líneas de montaje en concreto nos encontramos con dos tipos de programas autómatas:

- ✓ PL7-PRO: Software de la compañía francesa Telemecanique. En nuestro caso concreto usaremos la versión 4.4, que al tratarse de una de las más recientes va sobre sistema operativo Windows Xp. Los autómatas que utilizan este software son los: Premiun (TSXP57) y Micro (TSX37).

- ✓ PL7-3: Este software también pertenece a Telemecanique. Se trata de un producto más antiguo y por ello mucho menos dinámico. En este caso el sistema operativo utilizado es el OS2 (plataforma de IBM optimizada para equipos portátiles). En este caso los autómatas utilizados en las líneas con este tipo de programa son: TSX47 y TSX87.

En un capítulo posterior entraremos con más detalles a describir las características de los tipos de autómatas utilizados.

A continuación pasaremos a describir cómo hemos extraído la información distinguiendo entre las distintas formas que tienen estos programas PLC de acceder a la memoria de los palets a través del Inductel. Estas formas son:

- ✓ A través de la función WRITE_VAR Y READ_VAR del propio PL7.
- ✓ A través de la función SEND_REQ del propio PL7.
- ✓ Rutina estándar utilizada en la factoría: programada por los propios automatístas de Renault, y a la que la mayoría de los proveedores de programas PLC deben adaptarse. Con esto se consigue que una persona que no sepa como se ha programado esta función pueda utilizarla sin más que pasarle una serie de parámetros preestablecidos. Esta función internamente utiliza las funciones WRITE_VAR Y READ_VAR, pero antes de llamar a una de estas funciones hace una serie de operaciones y comprobaciones.

En los siguientes apartados entraremos a detallar cada una de estas formas de acceder al Inductel, además de hacer una introducción al lenguaje de contactos y al lenguaje literal estructurado.

2.2 INTRODUCCIÓN AL LENGUAJE DE CONTACTOS

Una sección de programa escrita en lenguaje de contactos está constituida por una serie de redes de contactos ejecutados secuencialmente por el autómata. La representación de una red de contactos es muy parecida a la de un esquema eléctrico. Las partes más importantes de la ventana en la que se programa el lenguaje de contactos son:

1. **Etiquetas**: variable de una red de contactos (opcional).
2. **Comentarios**: información sobre una red de contactos (opcional).
3. **Elementos gráficos**: las entradas/salidas del autómata (botones-pulsadores, detectores, relés, indicadores...), las funciones de los automatismos (temporizadores, contadores...), las operaciones aritméticas, lógicas y específicas y las variables internas del autómata.

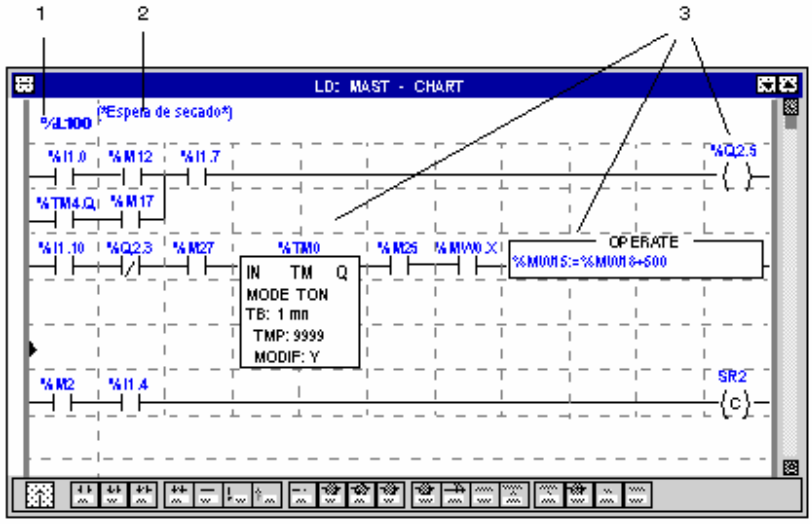


Figura 2.2: Partes más importantes de una ventana del lenguaje de contactos.

Una red de contactos se establece entre dos barreas de potencial, de forma que el sentido de circulación de la corriente se toma desde la barra de potencial izquierda hacia la barra de potencial derecha. La red se reparte en dos zonas:

- ✓ La zona de prueba, en la que constan las condiciones necesarias para una acción
- ✓ La zona de acción, que aplica el resultado consecutivo a un encadenamiento de prueba.

La etiqueta permite identificar una red en una entidad de programa (programa principal, subprograma,...) y suelen seguir la siguiente sintaxis: %Li con i comprendido entre 0 y 999. Además se suelen situar en la parte superior izquierda delante de la barra de potencial. Una variable de etiqueta sólo puede asignarse a una única red en el seno de una misma entidad de programa.

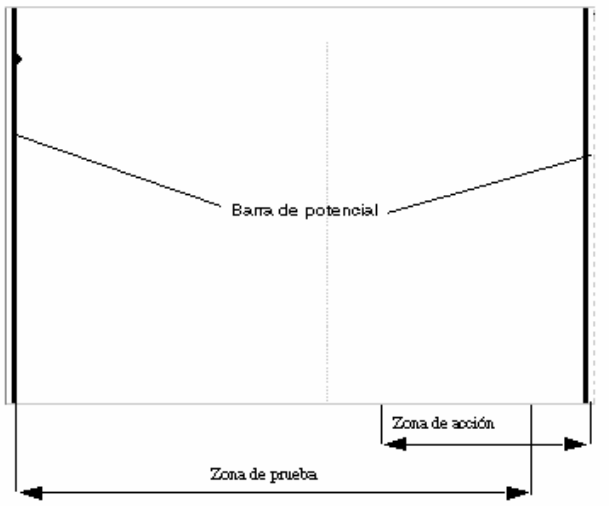


Figura 2.3: Red de contactos.

Los elementos gráficos son las instrucciones de una red de contactos. Existen los siguientes tipos:

- ✓ **Contactos,** se programan en la zona de prueba y ocupan una celda (1 línea de alto y una columna de ancho).

    1. Contacto de cierre: contacto de paso cuando el objeto bit que lo dirige está en el estado 1.
    2. Contacto de apertura: contacto de paso cuando el objeto bit que lo dirige está en el estado 0.
    3. Contacto de detección del flanco ascendente: paso de 0 a 1 del objeto bit que lo controla.
    4. Contacto de detección del flanco descendente: paso de 1 a 0 del objeto bit que lo controla.

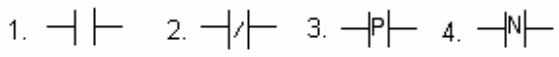


Figura 2.4: Contactos.

- ✓ **Elementos de unión,** permiten vincular los elementos gráficos de prueba y de acción.

    1. Conexión horizontal: permite vincular en serie los elementos gráficos de prueba y de acción entre las dos barras de potencial.
    2. Conexión vertical de potencial: permite vincular en paralelo los elementos gráficos de prueba y de acción.
    3. Derivación de cortocircuito: permite enlazar 2 objetos a través de varias conexiones.

Figura 2.5: Elementos de unión.

- ✓ **Bobinas,** se programan en la zona de acción y ocupan una celda (1 línea de alto y una columna de ancho).

    1. Bobina directa: el objeto bit asociado toma el valor del resultado de la zona de prueba.
    2. Bobina inversa: el objeto bit asociado toma el valor inverso del resultado de la zona de prueba.
    3. Bobina de conexión: el objeto bit asociado se sitúa en 1 cuando el resultado de la zona de prueba está en 1.
    4. Bobina de desconexión: el objeto bit asociado se sitúa en 0 cuando el resultado de la zona de prueba está en 1.
    5. Salto condicional a otra red (JUMP) : permite una ramificación hacia una red etiquetada, hacia arriba ó hacia abajo.
    6. Regreso del subprograma: reservado a los subprogramas SR, permite el regreso al módulo que llama cuando el resultado de la zona de prueba está en 1.
    7. Parada del programa: provoca la parada de la ejecución del programa cuando el resultado de la zona de prueba está a 1.

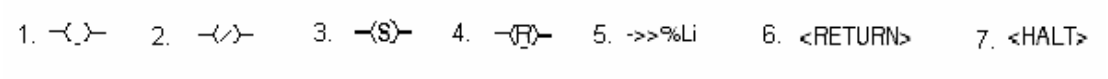


Figura 2.6: Bobinas.

- ✓ **Otros bloques funcionales programables.**

Las reglas principales a seguir para programar una red de contactos son:

- ✓ Los elementos gráficos simples de prueba y de acción ocupan cada uno una celda en el seno de una red.

- ✓ Cualquier línea de contactos empieza en la línea de potencial izquierda y debe terminar en la línea de potencial derecha.

- ✓ Las pruebas se sitúan siempre en las columnas de la 1 a la 10.Las acciones siempre se colocan en la columna 11.

- ✓ El sentido de circulación de la corriente es el siguiente:
    i. para las conexiones horizontales,
    ii. de la izquierda hacia la derecha, para las conexiones verticales, en ambos sentidos.

2.3 INTRODUCCIÓN AL LENGUAJE DE LITERAL ESTRUCTURADO

El lenguaje literal estructurado es un lenguaje del tipo algorítmico adaptado especialmente a la programación de funciones aritméticas complejas, manipulaciones de tablas y gestiones de mensajes.

Una sección de programa literal se organiza en frases, una frase literal equivale a una red de contactos en lenguaje de contactos.

Cada una de las frases, que empieza con un signo de exclamación (que se establece automáticamente), incluye los elementos siguientes:

1. Etiqueta: identifica una frase.
2. Comentario: muestra una frase.
3. Instrucciones: de una a varias instrucciones separadas por ";".

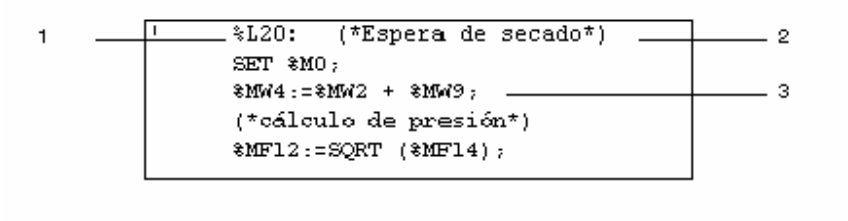


Figura 2.7: Frase.

El comentario puede integrarse en cualquier lugar de la frase, siendo ilimitado el número de comentarios por cada frase. Un comentario se enmarca por una parte y por otra con los caracteres (* y *).

Con el lenguaje lineal estructurado podemos utilizar un gran repertorio de posibilidades para componer las instrucciones:

- ✓ Estructuras de control: if, while, for, …
- ✓ Operaciones aritméticas y lógicas.
- ✓ Instrucciones sobre bits y tabla de bits.
- ✓ Instrucciones sobre cadenas.

La ejecución de un programa literal se efectúa de forma secuencial, instrucción por instrucción respetando las estructuras de control. En el caso de expresiones aritméticas o booleanas constituidas por varios operadores, existen una serie de reglas de prioridad.

2.4 ACCESO A TRAVÉS DE LA RUTINA ESTANDAR RENAULT

La mayoría de los programas PLC de las líneas bajo estudio acceden a la memoria del palet mediante esta forma. Esta rutina es usada tanto en con programas PL7-PRO y PL7-3. Necesita de tres parámetros que son:

- ✓ Dirección inicial de la memoria del palet, ya sea para escritura (primera posición donde se empieza a escribir) o para lectura (primera posición a leer).

- ✓ Longitud, cantidad de palabras a leer o escribir.

- ✓ Dirección inicial de la memoria del autómata, ya sea para escritura (primera posición donde se empieza a escribir) o para lectura (primera posición a leer).

La única distinción entre PL7-PRO y PL7-3 para el uso de esta función, es que PL7-3 trabaja con bytes y PL7-PRO lo hace con palabras equivalente a dos bytes) para los dos primeros parámetros, mientras que para el último sigue trabajando con palabras. Para que todo sea coherente los autómatas en PL7-3 deben poner el primer parámetro a un valor impar y el segundo a uno par.

Como ya comentamos anteriormente las rutinas estándar de Renault se basan en una serie de comprobaciones previas para finalmente acabar haciendo una llamada a la función de biblioteca de PL7, que en este caso serán WRITE_VAR Y READ_VAR.

La forma de utilizar esta rutina es la siguiente:

- ✓ En la tabla de constantes del PLC se escriben los tres parámetros (dirección en Inductel, longitud y dirección en PLC) de todos los accesos que realice el programa a través del Inductel.

- ✓ Cuando dentro del programa autómata se realice (mediante la activación de la correspondiente bobina) una petición de lectura o de escritura, se procederá a la ejecución la rutina estándar

- ✓ La rutina estándar lee los tres parámetros asociados a ese bit de petición de la tabla de constantes y directamente pasa a ejecutarse. Las funciones estándar de lectura y escritura están programadas en lenguaje literal estructurado.

A continuación pasaremos a ver dos ejemplos de tablas de constantes, la primera en el que el autómata solo controla un Inductel y la segunda en la que controla varios. Para ilustrarlo de forma más gráfica usaremos ilustraciones sacadas directamente del programa PL7-PRO.

En este primer caso mostramos la tabla de constantes de un programa PLC que controla un solo Inductel y que realiza dos lecturas. La primera lectura es de tamaño dos y las palabras leídas son las posiciones 126 y 127 del Mapping, que significan respectivamente: pieza buena o mala; y última puesto que trabajó sobre la pieza. La segunda lectura es de tamaño uno, con lo cual solo lee la palabra 50, donde se guarda en el primer byte un código relacionado con la caja diferencial y en el segundo otro código relacionado con el cárter de embrague.

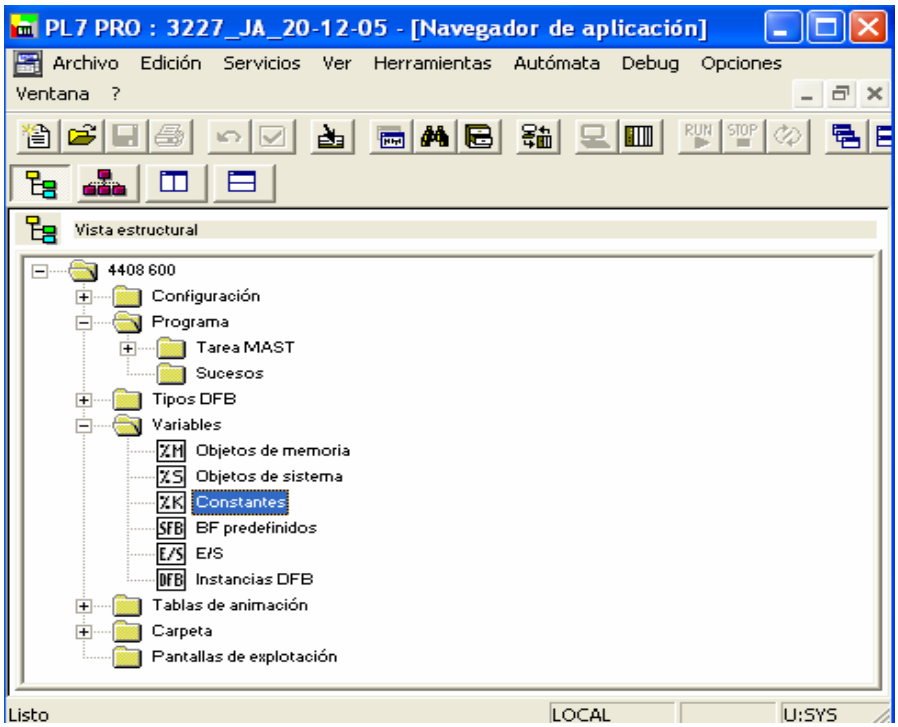
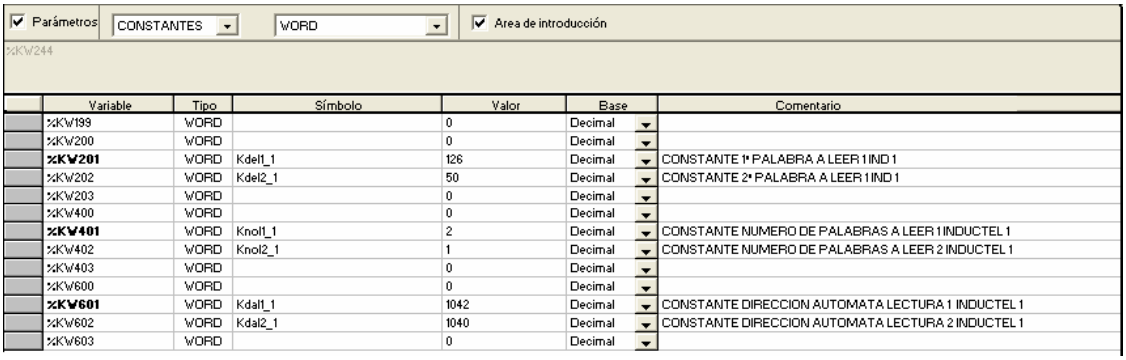


Figura 2.8: Navegador de aplicación PL7-PRO.


Parámetros CONSTANTES WORD Area de introducción

%KW244

	Variable	Tipo	Símbolo	Valor	Base	Comentario
	%KW199	WORD		0	Decimal	
	%KW200	WORD		0	Decimal	
	%KW201	WORD	Kdel1_1	126	Decimal	CONSTANTE 1ª PALABRA A LEER 1 IND 1
	%KW202	WORD	Kdel2_1	50	Decimal	CONSTANTE 2ª PALABRA A LEER 1 IND 1
	%KW203	WORD		0	Decimal	
	%KW400	WORD		0	Decimal	
	%KW401	WORD	Knol1_1	2	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 1 INDUCTEL 1
	%KW402	WORD	Knol2_1	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 2 INDUCTEL 1
	%KW403	WORD		0	Decimal	
	%KW600	WORD		0	Decimal	
	%KW601	WORD	Kdal1_1	1042	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 1 INDUCTEL 1
	%KW602	WORD	Kdal2_1	1040	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 2 INDUCTEL 1
	%KW603	WORD		0	Decimal	

Figura 2.9: Ejemplo 1.

En el siguiente ejemplo tenemos el caso de un PLC que controla varios Inducteles, y para cado uno de ellos realiza varias lecturas y escrituras. A diferencia de los PLCs que

controlan un solo Inductel, ahora se nos plantea una duda: ¿Qué número de puesto corresponde a cada Inductel? Para solucionar cuestión nos apoyamos en que la gran mayoría de los puestos realizan lo que comúnmente conocemos como "firma". La firma no consiste más que en escribir su número de puesto en la palabra 127 del Mapping (Esa posición del Mapping es usada para almacenar el número del último puesto en trabajar sobre la pieza). Para hacer esta comprobación buscamos mediante las referencias cruzadas el segmento de lenguaje de contacto donde se escribe en la palabra 127 y vemos que valor se guarda en esa posición (ese valor será el número de puesto).

Para este segundo caso de nuevo solo representamos las lecturas, ya que las constantes para escritura son análogas. Ahora el primer Inductel realiza seis lecturas mientras que el segundo y el tercero realizan cinco.

Parámetros: CONSTANTES | WORD | Area de introducción

%KW245

Variable	Tipo	Símbolo	Valor	Base	Comentario
%KW200	WORD		0	Decimal	
%KW201	WORD	Kdel1_1	126	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 1 INDUCTEL 1
%KW202	WORD	Kdel2_1	39	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 2 INDUCTEL 1
%KW203	WORD	Kdel3_1	130	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 3 INDUCTEL 1
%KW204	WORD	Kdel4_1	50	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 4 INDUCTEL 1
%KW205	WORD	Kdel5_1	55	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 5 INDUCTEL 1 FORZADO 2 OCTETO 110
%KW206	WORD	Kdel6_1	40	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 5 INDUCTEL 1 FORZADO 2 OCTETO 110
%KW207	WORD		0	Decimal	
%KW211	WORD	Kdel1_2	126	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 1 INDUCTEL 2
%KW212	WORD	Kdel2_2	39	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 2 INDUCTEL 2
%KW213	WORD	Kdel3_2	50	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 3 INDUCTEL 2
%KW214	WORD	Kdel4_2	130	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 4 INDUCTEL 2
%KW215	WORD	Kdel5_2	251	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 5 INDUCTEL 2
%KW220	WORD		0	Decimal	
%KW221	WORD	Kdel1_3	126	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 1 INDUCTEL 3
%KW222	WORD	Kdel2_3	39	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 2 INDUCTEL 3
%KW223	WORD	Kdel3_3	50	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 3 INDUCTEL 3
%KW224	WORD	Kdel4_3	130	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 4 INDUCTEL 3
%KW225	WORD	Kdel5_3	251	Decimal	CONSTANTE DIRECCION INDUCTEL LECTURA 5 INDUCTEL 3
%KW226	WORD		0	Decimal	
%KW401	WORD	Knol1_1	2	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 1 INDUCTEL 1
%KW402	WORD	Knol2_1	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 2 INDUCTEL 1
%KW403	WORD	Knol3_1	2	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 3 INDUCTEL 1
%KW404	WORD	Knol4_1	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 4 INDUCTEL 1
%KW405	WORD	Knol5_1	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 5 INDUCTEL 1 FORZADO A 2 OCTETO 110
%KW406	WORD	Knol6_1	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 5 INDUCTEL 1 FORZADO A 2 OCTETO 110
%KW407	WORD		0	Decimal	
%KW411	WORD	Knol1_2	2	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 1 INDUCTEL 2
%KW412	WORD	Knol2_2	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 2 INDUCTEL 2
%KW413	WORD	Knol3_2	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 3 INDUCTEL 2
%KW414	WORD	Knol4_2	2	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 4 INDUCTEL 2
%KW415	WORD	Knol5_2	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 5 INDUCTEL 2
%KW416	WORD		0	Decimal	
%KW421	WORD	Knol1_3	2	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 1 INDUCTEL 3
%KW422	WORD	Knol2_3	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 2 INDUCTEL 3
%KW423	WORD	Knol3_3	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 3 INDUCTEL 3
%KW424	WORD	Knol4_3	2	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 4 INDUCTEL 3
%KW425	WORD	Knol5_3	1	Decimal	CONSTANTE NUMERO DE PALABRAS A LEER 5 INDUCTEL 3
%KW600	WORD		0	Decimal	
%KW601	WORD	Kdal1_1	260	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 1 INDUCTEL 1
%KW602	WORD	Kdal2_1	262	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 2 INDUCTEL 1
%KW603	WORD	Kdal3_1	264	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 3 INDUCTEL 1
%KW604	WORD	Kdal4_1	263	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 4 INDUCTEL 1
%KW605	WORD	Kdal5_1	6666	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 5 INDUCTEL 1 FORZADO 2 OCTETO 110
%KW606	WORD	Kdal6_1	6668	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 5 INDUCTEL 1 FORZADO 2 OCTETO 110
%KW610	WORD		0	Decimal	
%KW611	WORD	Kdal1_2	276	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 1 INDUCTEL 2
%KW612	WORD	Kdal2_2	278	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 2 INDUCTEL 2
%KW613	WORD	Kdal3_2	279	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 3 INDUCTEL 2
%KW614	WORD	Kdal4_2	280	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 4 INDUCTEL 2
%KW615	WORD	Kdal5_2	298	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 5 INDUCTEL 2
%KW620	WORD		0	Decimal	
%KW621	WORD	Kdal1_3	292	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 1 INDUCTEL 3
%KW622	WORD	Kdal2_3	294	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 2 INDUCTEL 3
%KW623	WORD	Kdal3_3	295	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 3 INDUCTEL 3
%KW624	WORD	Kdal4_3	296	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 4 INDUCTEL 3
%KW625	WORD	Kdal5_3	299	Decimal	CONSTANTE DIRECCION AUTOMATA LECTURA 5 INDUCTEL 3

Figura 2.10: Ejemplo 2.

El código asociado a las rutinas estándar de Renault como ya se comento anteriormente está basado en lenguaje literal estructurado, y es el siguiente:

✓ Función de lectura:

```
!
(* Control de Sobrepasamiento de Puntero *)
IF Pun_pet>199 OR Pun_pet<0 THEN (* PUNTERO DE PETICIÓN *)
        Pun_pet:=0;
END_IF;
IF Pun_int>5 THEN (* PUNTERO Nº INTENTOS TRAS DEFECTO *)
        Pun_int:=0;
END_IF;
!
(* Búsqueda de Petición o Error de Lectura *)
WHILE((NOT Pl1_1[Pun_pet])OR(Dl1_1[Pun_pet]))AND(Pun_pet<200)DO
        INC Pun_pet;
END_WHILE;
!
(* Petición Encontrada, Carga Valores de Inductel *)
IF Pl1_1[Pun_pet] AND(NOT Dl1_1[Pun_pet])THEN
        In_r_1_ini_1:=Kdel1_1[Pun_pet];
        In_r_long:=Knol1_1[Pun_pet];
        In_r_pos:=Kdal1_1[Pun_pet];
        In_r_dir_induc:6:=ADR#0.0;
        SET In_r_pet;
       (* Calcula Nº de Inductel en Funcion de Peticion *)
        In_r_op_a:=Krl1_1[Pun_pet];
        In_r_op_b:=SHL(In_r_op_a,8);
        In_r_op_c:=In_r_op_b+16#0000;
END_IF;
!
(* LECTURA *)
In_r_var:=Sys_read_var;(* IMAGEN DEL BIT DE ACTIVIDAD READ_VAR *)
(* Lectura Realizada Compara Respuesta *)
IF FE In_r_var THEN
(* Si Lectura Errónea, Defecto Lectura al 5º Intento *)
IF  In_r_operation<>0  THEN (* %MW211 OPERACIÓN. <>0 INCORRECTO; =0
CORRECTO *)
        IF Pun_int>4 THEN (*  5º INTENTO DE ESCRITURA, DEFECTO PETICIÓN *)
                SET Dl1_1[Pun_pet];
                RESET Pl1_1[Pun_pet];
                RESET In_r_pet;
        ELSE
                INC Pun_int;
        END_IF;
END_IF;
```

```
(* Si lectura Correcta, Memoria Lectura *)
        IF In_r_operation=0 THEN
                SET Ml1_1[Pun_pet];
                RESET Pl1_1[Pun_pet];
                RESET In_r_pet;
                Pun_int:=0;
        END_IF;
END_IF;

(* Lanzamiento Lectura *)
IF In_r_pet AND NOT Sys_read_var THEN
        In_r_valor:=5;
        READ_VAR(In_r_dir_induc:6,'%MW',In_r_1_ini_1,In_r_long,Genera[I
n_r_pos]:10,In_read_var:4);
END_IF;
In_r_var:=Sys_read_var;
```

✓ Función de escritura:

```
!
 (* Control de Sobrepasamiento de Puntero *)
IF %MW9300>199 OR %MW9300<0 THEN (* PUNTERO DE PETICIÓN *)
   %MW9300:=0;
END_IF;
IF %MW9301>5 THEN (* PUNTERO Nº INTENTOS TRAS DEFECTO *)
   %MW9301:=0;
END_IF;
!
(* Búsqueda de Petición o Error de Escritura *)
WHILE((NOT
Pe1_1[%MW9300])OR(De1_1[%MW9300]))AND(%MW9300<200)DO
  INC %MW9300;
END_WHILE;
!
(* Petición Encontrada, Carga Valores de Inductel *)
IF Pe1_1[%MW9300] AND(NOT De1_1[%MW9300])THEN
   In_w_1_ini_1:=Kdee1_1[%MW9300];
   In_w_long:=Knoe1_1[%MW9300];
   In_w_pos:=Kdae1_1[%MW9300];
   In_w_dir_induc:6:=ADR#0.0;
   SET In_w_pet;
(* Calcula Nº de Inductel en Función de Petición *)
   In_w_op_a:=Kwl1_1[%MW9300];
   In_w_op_b:=SHL(In_w_op_a,8);
   In_w_op_c:=In_w_op_b+16#0000;
END_IF;
```

```
!
(* ESCRITURA *)
In_w_var:=In_write_var:X0;(* IMAGEN DEL BIT DE ACTIVIDAD WRITE_VAR *)

(* Escritura Realizada Compara Respuesta *)
IF FE In_w_var THEN

(* Si Escritura Errónea, Defecto Escritura *)
  IF In_w_operation<>0 THEN (* %MW411 OPERACIÓN. <>0
INCORRECTO ;=0 CORRECTO *)

        IF %MW9301>4 THEN (* AL 5º INTENTO DE ESCRITURA
DEFECTO PETICIÓN *)
                SET De1_1[%MW9300];
                RESET Pe1_1[%MW9300];
                RESET In_w_pet;
        ELSE
                INC %MW9301;
        END_IF;
  END_IF;

(* Si Escritura Correcta, Memoria Escritura *)
  IF In_w_operation=0 THEN

        SET Me1_1[%MW9300];
        RESET Pe1_1[%MW9300];
        RESET In_w_pet;
        %MW9301:=0;
  END_IF;
END_IF;

(* Lanzamiento Escritura *)
IF In_w_pet AND NOT In_write_var:X0 THEN
  In_w_valor:=5;
  WRITE_VAR(In_w_dir_induc:6,'%MW',In_w_1_ini_1,In_w_long,Gene
  ra[In_w_pos]:10,In_write_var:4);
END_IF;
In_w_var:=In_write_var:X0;
```

2.5 ACCESO ATRAVES DE WRITE-VAR Y READ-VAR

WRITE_VAR Y READ_VAR son dos funciones de biblioteca de PL7-PRO y PL7-3 específicas para comunicaciones con Inducteles y basadas en el estándar CANopen. La forma de usarlas consiste en colocarlas directamente en la parte de acción de cualquiera de los segmentos, dentro de un bloque de operación.

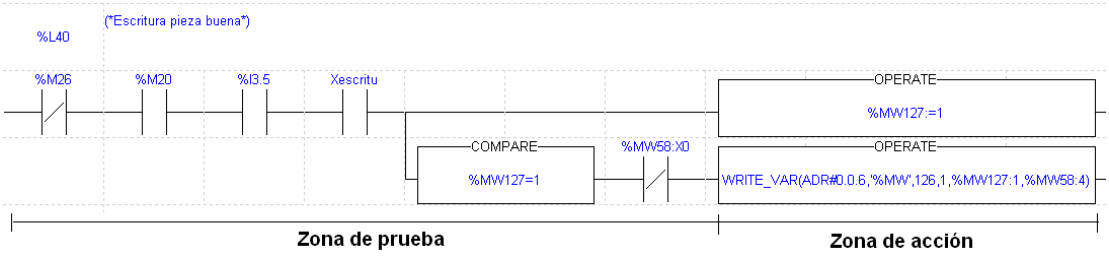


Figura 2.11: Ejemplo con WRITE_VAR.

Los parámetros que utilizan se especifican a continuación:

- ✓ WRITE_VAR(Inductel, `SDO', gestión, cantidad, %MWi, %MWk)

    1. Inductel: Dirección de la entidad de destino del intercambio, dirección del Inductel.
    2. SDO: Para hacer el acceso conforme a es estándar CANopen.
    3. Gestión: forma de realizar la gestión de errores.
    4. Cantidad: longitud de la escritura a realizar.
    5. %MWi: dirección inicial de la memoria del Inductel.
    6. %MWk: dirección inicial de la memoria del autómata.

- ✓ READ_VAR(Inductel, 'SDO', gestión, cantidad, %MWi, %MWk)

    1. Inductel: Dirección de la entidad de destino del intercambio, dirección del Inductel.
    2. SDO: Para hacer el acceso conforme a es estándar CANopen.
    3. Gestión: forma de realizar la gestión de errores.
    4. Cantidad: longitud de la lectura a realizar.
    5. %MWi: dirección inicial de la memoria del Inductel.
    6. %MWk: dirección inicial de la memoria del autómata.

2.6 ACCESO A TRAVÉS DE SEND-REQ

De nuevo SEND_REQ es una función de biblioteca de PL7-PRO y PL7-3 específica para comunicaciones con Inducteles. La función SEND_REQ permite la codificación y la emisión de todas las peticiones UNI-TE y Modbus/Jbus, así como la recepción de las respuestas asociadas. La forma de usarlas consiste en colocarlas directamente en la parte de acción de cualquiera de los segmentos, dentro de un bloque de operación.

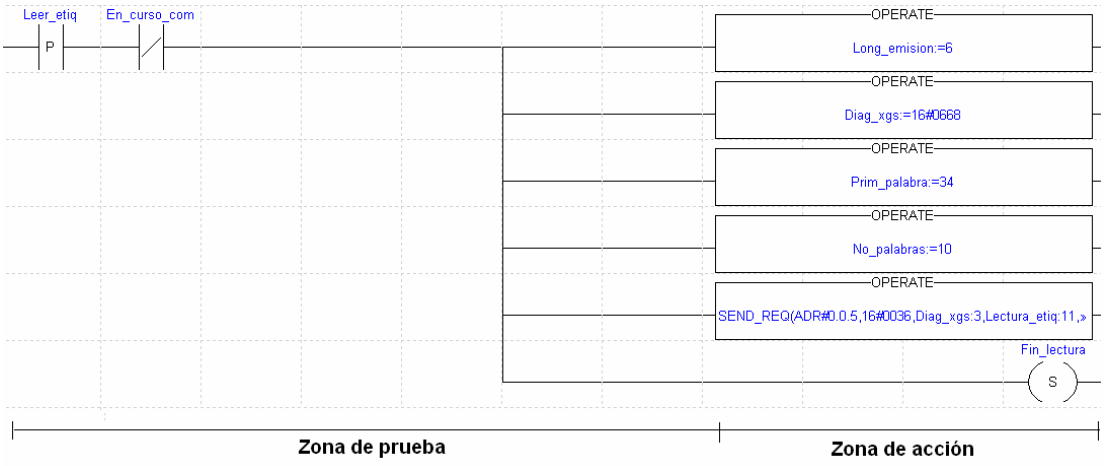


Figura 2.12: Ejemplo con SEND_REQ.

Los parámetros que utilizan se especifican a continuación:

SEND_REQ(Inductel, 'UNI-TE', cantidad, %MWi, %MWk)

1. Inductel: Dirección de la entidad de destino del intercambio, dirección del Inductel.

2. UNI-TE: este parámetro necesita el valor del código requerido conforme al estándar UNI-TE.

3. Cantidad: longitud de la escritura a realizar.

4. %MWi: dirección inicial de la memoria del Inductel.

5. %MWk: dirección inicial de la memoria del autómata.

2.7 ELABORACIÓN DE LAS FICHAS DE PUESTO

2.7.1 INTRODUCCIÓN

La ficha de puesto consiste en la unidad básica de información para este proyecto. Está basada en una hoja de cálculo Excel con un formato determinado, en el que se almacena todos los datos relacionados con el acceso a la pastilla de memoria de cada uno de los puestos de lectura/escritura de los Inducteles.

El formato de las fichas de puesto usado en el proyecto ha sido modificado respecto al que se utilizaba previamente en la factoría. Las razones que motivan este cambio son varias deficiencias que presentan las antiguas fichas de puesto, como por ejemplo;

- ✓ Las palabras o bytes accedidos tanto en lectura como en escritura se indicaban en forma de rango, de forma que puede que alguna palabra intermedias entre los dos extremos indicados no fuera utilizada por el programa autómata, pareciendo de esta forma que sí lo hiciese.

- ✓ No existe información alguna en las que se indique el lugar o zona de la nave de montaje en la que se encuentra dicho puesto. Como compensación a esta deficiencia los ficheros con las fichas de puesto eran almacenados en una estructura de directorios cuya ruta indica la zona de la nave de montaje en la que se encuentra; para ello se le daban nombres de las zonas a los directorios de almacenamiento.

- ✓ No existía ningún campo para indicar cual era el autómata que controlaba dicho puesto de lectura o escritura.

Si a esta serie de problemas les unimos que las fichas no estaban actualizadas (debido a la gran cantidad de modificaciones que se producen en los programas autómatas) queda claro el porqué se tomó la decisión de crear unas nuevas fichas de puesto partiendo de cero; tanto en un nuevo formato, como en la información del Mapping.

2.7.2 NUEVO FORMATO DE LAS FICHAS DE PUESTO

En este apartado del capítulo se describirán los distintos campos que componen el nuevo formato de las fichas de puesto. Dichos campos son los siguientes:

- ✓ **Localización,** este campo nos permite situar el lugar donde esta el puesto de lectura/escritura dentro de la nave de montaje. A su vez se subdivide en varias partes:
    - Zona, conjunto de máquinas que se agrupan debido a alguna característica común, como puede ser su cometido.

- Máquina, en la que el Inductel está ubicado. Se especifica con la matricula de la máquina, que consiste en una serie de cuatro dígitos numéricos.
- Puesto, número de puesto asignado a cada inductel. Dicho número es siempre inferior a tres cifras.

✓ **Fecha,** fecha de la última modificación sufrida por la ficha.

✓ **Versión,** la cual se debe ir modificando con las sucesivas actualizaciones realizadas a las fichas de puesto.

✓ **Fichero,** nombre del fichero del programa autómata PLC que controla dicho puesto. Son de extensión *stx* en caso de que se utilice el software PL7-PRO. Para el caso de software PL7-3 no se puede ofrecer el nombre y por ello tan solo se indica que se utiliza dicho software.

✓ **Palabras o bytes accedidos en lectura,** se ha reservado la primera columna para indicar todas las palabras del Mapping accedidas en lectura. Las palabras se especifican una a una en cada celda de esta primera columna.

✓ **Palabras o bytes accedidos en escritura,** en este caso la columna habilitada es la tercera.

✓ **Comentarios,** se pueden hacer comentarios sobre cada una de las palabras, para ello se habilitan la segunda (para la lectura) y cuarta columna (para la escritura).

✓ **Tablas de direcciones,** la forman tres columnas: dirección memoria Mapping, cantidad y dirección memoria autómata. Hay dos tablas una para los accesos en escritura y otra para la lectura. Estas dos tablas son una petición expresa de los automatistas de la factoría, ya que facilitan su trabajo. Los datos de estas tablas son extraídos durante las revisiones de los programas PLC.

✓ **Datos adicionales,** en la parte inferior de la hoja queda un espacio sobrante que se ha habilitado para que los automatistas puedan añadir información acerca del funcionamiento del programa PLC que controla dicho puesto. Esta información consiste en la forma de proceder del autómata en función de los valores leídos en cada palabra, además de los valores a escribir en cada uno de los estados o casos en los que nos podamos encontrar.

2.7.3 NORMAS PARA LA CONSISTENCIA DE LAS FICHAS DE PUESTO

Las fichas deben cumplir una serie de requisitos para asegurarnos que son consistentes para su interacción con las macros de la interfaz. Dichas reglas son:

- ✓ Las palabras deben empezar por W (en caso de palabras) o B (en caso de bytes), seguido de un número inferior a 610 (en el caso de palabras) o 1210 (en el caso de bytes), correspondientes a los limites del mapa de memoria del Mapping.

- ✓ Debemos introducir la primera palabra en la primera celda habilitada (fila 7) y seguir hacia abajo sin dejar ninguna celda vacía o en blanco, hasta que terminemos con todas las palabras.

- ✓ Las fichas deben nombrarse de la siguiente forma:

    - "Puesto nº", en caso de que exista un único puesto con ese número de puesto asignado. El número será de tres o menos cifras.

    - "Puesto nº (nº de máquina)", en caso de que exista más de un puesto de lectura/escritura con el mismo número de puesto. De nuevo el número debe tener tres o menos cifras y el nº de máquina debe tener cuatro de forma obligatoria.

A continuación mostramos la imagen de una ficha de puesto para hacernos una idea más detallada de los campos comentados anteriormente:

1	FACTORÍA RENAULT SEVILLA													
2	Ficha de puesto para el Mapping													
3	LOCALIZACIÓN: Linea 1 Dosificación Loctite Zona B1 MAQUINA: 2689 PUESTO: 94													
4	Fecha: 2/05/2006			FICHERO: loctite.stx							Versión: v 1.1			
5	PALABRAS ACCEDIDAS EN LECTURA						PALABRAS ACCEDIDAS EN ESCRITURA							
6	PALABRA O BYTE		COMENTARIO				PALABRA O BYTE		VALOR ESCRITO			COMENTARIO		
7	W252						B252							
8	W127						W127							
9	B101						W269							
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														
32														
33														
34														
35	TABLA DE CONSTANTES EN LECTURA								TABLA DE CONSTANTES EN ESCRITURA					
36	PRIMER OCTETO MAPPING		CATIDAD		PRIMER OCTETO AUTOMATA				PRIMER OCTETO MAPPING		CANTIDAD		PRIMER OCTETO AUTOMATA	
37	W126		2		W500				W126		2		W1000	
38	W50		1		W502				W269		1		W1002	
39									W0		1		W1004	
40														
41														
42														

Figura 2.13: Ficha de puesto.