

Direccionamiento en STEP 7

Automatización Industrial

Índice

TEMA – Direccionamiento en STEP7

1. – Tipos de Direccionamiento
 - 1.1 – *Direccionamiento Inmediato*
 - 1.2 – *Direccionamiento Directo*
 - 1.3 – *Direccionamiento Indirecto*
2. – Punteros y Tipos de Punteros
 - 2.1 – *Punteros de Número*
 - 2.2 – *Punteros a Área*
 - 2.3 – *Punteros a DB*
 - 2.4 – *Punteros ANY*
3. – Direccionamiento Indirecto por Memoria
4. – Direccionamiento Indirecto por Registro
5. – Operaciones Relacionadas con Registros

Tipos de Direccionamiento

Tipos de Direccionamiento

- Direccionamiento Inmediato
- Direccionamiento Directo
- Direccionamiento Indirecto

Direccionamiento Inmediato

El valor del operando está codificado directamente en la operación
Ejemplo: L W#16#3478

Direccionamiento Directo

La dirección del operando se encuentra codificada en la operación
Ejemplo: L MW 122; A I 1.2

Direccionamiento Indirecto

El operando indica la dirección del valor que va a procesar la operación.
Ejemplo: A I [MD 2];

Direccionamiento Inmediato

Direccionamiento Inmediato

El valor del operando está codificado directamente en la operación, es decir que la operación va seguida directamente del operando con el que operará. Por otra parte, una operación también puede aportar su propio valor.

Ejemplo	Descripción
SET	Poner el RLO a 1.
OW W#16#A320	O con palabras.
L 27	Cargar el valor entero 27 en el ACU 1.
L 'ABCD'	Cargar los caracteres ASCII ABCD en el ACU 1.
L B#(100,12)	Cargar una constante de 2 bytes (100 y 12) en el ACU 1.
L C#0100	Cargar el valor BCD 100 en el ACU 1.

Direccionamiento Directo

Direccionamiento Directo

La dirección del operando se encuentra codificada en la operación, es decir, el operando indica la dirección del valor que va a procesar la operación. El operando se compone de dos partes:

- un identificador (p. ej. “IB” para “byte de entrada”)
- una dirección exacta dentro del área de memoria indicada por el identificador.

El operando indica directamente la dirección del valor.

Ejemplo	Descripción
A I 0.0	Ejecutar una operación lógica Y con el bit de entrada I 0.0.
S L 20.0	Activar el bit de datos locales L20.0.
= M 115.4	Asignar el RLO al bit de marcas M115.4.
L IB 0	Cargar el byte de entrada IB0 en el ACU 1.
L MW 64	Cargar la palabra de marcas MW64 en el ACU 1.
T DBD 12	Transferir el contenido del ACU 1 a la palabra doble de datos DBD12.

Direccionamiento Directo

Área	Ejemplo	Tipo de Acceso	Descripción
I	I 0.0	Bit, Byte, Palabra, Doble palabra	Imagen de Proceso de Entradas
Q	QB 4	Bit, Byte, Palabra, Doble palabra	Imagen de Proceso de Salidas
PI	PI W 10	Byte, Palabra, Doble palabra	Entradas de periferia
PQ	PQ D 24	Byte, Palabra, Doble palabra	Salidas de periferia
M	M 100.0	Bit, Byte, Palabra, Doble palabra	Marcas
T	T 15	-	Temporizadores
C	C 80	-	Contadores
DB	DBX0.0	Bit, Byte, Palabra, Doble palabra	Datos de un DB global
DI	DIW20	Bit, Byte, Palabra, Doble palabra	Datos de un DB de instancia
L	LB 34	Bit, Byte, Palabra, Doble palabra	Datos locales

Direccionamiento Indirecto

Direccionamiento Indirecto

- Permite asignar direcciones que no son conocidas hasta la ejecución
- Permite múltiples procesamientos de partes del programa
- Como las direcciones no son calculadas hasta la ejecución hay un riesgo de que áreas de la memoria sean sobrescritas.

Ejemplo	Descripción
A 1 [MD 2] ó A 1 ["punter"]	Ejecutar una combinación Y con el bit de entrada cuya dirección exacta se encuentra en la palabra doble de marcas MD2 o en la dirección denominada "punter" en la tabla de símbolos, como referencia a MD2.
= DIB [DBD2]	Asignar el bit RLO al bit de datos de instancia cuya dirección exacta se encuentra en la palabra doble de datos DBD2.
L IB [DID 4]	Cargar en el ACU 1 el byte de entrada cuya dirección exacta se encuentra en la palabra doble de datos de instancia DID4.
OPN DB [LW2]	Abrir el bloque de datos cuyo número se encuentra en la palabra de datos local LW2.
O Q [LD 3] u O Q ["punt2"]	Ejecutar una combinación O con el bit de salida cuya dirección exacta se encuentra en la palabra doble de datos locales 3 o en la dirección denominada "punt2" en la tabla de símbolos, como referencia a LD3.

Tipos de Direccionamiento Indirecto

Direcciones Indirectas

<i>Direcciones que pueden ser especificadas de Modo Indirecto</i>	<i>Direccionamiento</i>	<i>Puntero</i>
E/S de Periferia, Entradas, Salidas, Bits de Memoria, Datos Globales, Datos de Instancia, Datos Locales, Datos Temporales	Indirecto por Memoria Indirecto por Registro	Puntero de Área
Temporizadores (T), Contadores (C), Funciones (FC's), Bloques Funcionales (FB's), Boques de Datos (DB's)	Indirecto por Memoria	Número de 16 bits

Tipos de Direccionamiento Indirecto

- Direccionamiento Indirecto por Memoria con Número
- Direccionamiento Indirecto por Memoria con Puntero de Área
- Direccionamiento Indirecto por Registro e Intraárea (área interna)
- Direccionamiento Indirecto por Registro e Interárea (área cruzada)

Punteros y Tipos de Punteros

Punteros

Un puntero se usa para apuntar a una dirección

Tipos de Punteros

- Punteros a Área: son de 32 bits de longitud y contiene una dirección específica
- Punteros a DB: son de 48 bits de longitud y contienen el número de DB además del puntero a área
- Punteros ANY: son de 80 bits de longitud y contienen especificaciones adicionales (como tipo de dato de la dirección) además del puntero a DB

Punteros a Área (I)

Definición

El puntero a área contiene la dirección y es posible también el área direccionada:

Tipos de Punteros a Área

1. Intraárea (Área Interna): El puntero no especifica el área direccionada
2. Interárea (Área Cruzada): El puntero especifica el área direccionada

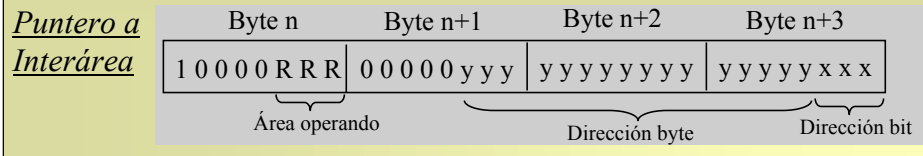
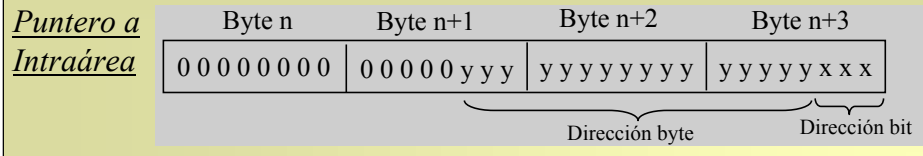
Notación Constante

1. Intraárea (Área Interna): **P#y.x** Ejemplo: P#22.0
2. Interárea (Área Cruzada): **P#Zy.x** Ejemplo: P#M22.0

- x=dirección bit; y=dirección byte, Z=área

Con el área se especifica el identificador de la dirección

Punteros a Área (II)



Código:

Bit 31=1 El registro contiene una dirección y la zona de memoria .
 Bit 31=0 El registro contiene una dirección y no la zona de memoria .

Zonas de acceso (Áreas de Operando):

RRR	Identificador:	000	Periferia
		001	Entradas (PII)
		010	Salidas (PIQ)
		011	Marcas
		100	Datos en DB Global (DB)
		101	Datos en DB de instancia (DI)
		110	Datos locales del módulo (L)
		111	Datos locales del módulo anterior (V)

Rango de direcciones:

Y...Y Dirección Byte
 XXX Dirección Bit

Punteros a DB

Definición

Un puntero a DB, además de un puntero a área, contiene un número de bloque de datos como un número positivo INT en adición al puntero a área.

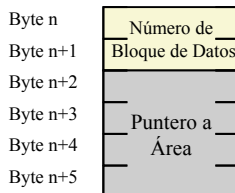
Se especifica el Bloque de Datos si el puntero de área contiene las áreas de dirección de DB global o DB de instancia. En los demás casos los dos primeros bytes son cero.

Notación Constante

P#DataBlock.DataAddress

Ejemplo: P#DB10.DBX 20.5

Puntero a DB



Punteros ANY (I)

Definición

Un puntero ANY, además de un puntero a DB, contiene un tipo de datos y un factor de repetición.

Tipos de Punteros ANY

1. *Variables con Tipos Datos:* El puntero contiene un puntero DB
2. *Variables con Tipos Parámetros:* El puntero contiene un número (temporizadores, Contadores y Bloques)

Notación Constante

1. *Variables con Tipos Datos:* **P#[Bloque de Datos.] Dirección Tipo Cantidad**
2. *Variables con Tipos Parámetros :* **L#Número Tipo Cantidad**

Ejemplos:

```
P#DB11.DBX 30.0 INT 12 ; Área con 12 palabras en DB11 desde DBB30
P#M 16.0 BYTE 8 ; Área con 8 Bytes desde MB16
P#I 18.0 WORD 1 ; Palabra de Entradas IW18
P#I 1.0 BOOL 1 ; Entrada I1.0
L#10 TIMER 1 ; Temporizador T10
L#2 COUNTER 1 ; Contador C2
```

Punteros ANY (II)

	<i>Puntero ANY para Tipos Datos</i>	<i>Puntero ANY para Temp/Cont</i>	<i>Puntero ANY para Bloques</i>	<u>Campo Tipo en el Puntero ANY</u>	
Byte n	16#10	16#10	16#10	Tipos de Datos Elementales 01 BOOL 02 BYTE 03 CHAR 04 WORD 05 INT 06 DWORD 07 DINT 08 REAL 09 DATE 0A TOD 0B TIME 0C S5TIME	Tipos de Datos Complejos 0E DT 13 STRING Tipos Parámetro 17 BLOCK_FB 18 BLOCK_FC 19 BLOCK_DB 1A BLOCK_SDB 1C COUNTER 1D TIMER
Byte n+1	Tipo	Tipo	Tipo		
Byte n+2	Cantidad	Cantidad	Cantidad		
Byte n+3					
Byte n+4	Número de Bloque de Datos	16#0000	16#0000		
Byte n+5					
Byte n+6		Tipo	16#0000		
Byte n+7	Puntero a Área	16#00			
Byte n+8		Número	Número		
Byte n+9					

Direccionamiento Indirecto por Memoria (I)

En el direccionamiento indirecto por memoria, el operando indica la dirección del valor que va a procesar la operación. El operando se compone de las siguientes partes:

- Un identificador (p. ej. "IB" para "byte de entrada")
- Una palabra que contiene el número de un temporizador (T), de un contador (Z), de un bloque de datos (DB), de una función (FC) o de un bloque de función (FB)
- Una palabra doble que contiene la dirección exacta de un valor del área de memoria, indicada por el identificador.

Si utiliza un operando indirecto por memoria que esté almacenado en el área de memoria del bloque de datos, deberá abrir primero el bloque de datos, utilizando a tal efecto la operación *Abrir bloque de datos*.

Ejemplo: OPN DB10
 L IB [DBD20]

Direccionamiento Indirecto por Memoria (II)

El operando indica la dirección del valor o del número de forma indirecta, es decir, utilizando el puntero. Esta palabra o palabra doble puede encontrarse en una de las siguientes áreas:

- Marca (M)
- Bloque de datos (DB)
- Bloque de datos de instancia (DI)
- Datos locales (L).

Ejemplo	Descripción
A I [MD 2] ó A I [ana]	Ejecutar una combinación Y con el bit de entrada cuya dirección exacta se encuentra en la palabra doble de marcas MD2 o en la dirección denominada "Ana" en la tabla de símbolos, como referencia a MD2.
= DIB [DBD2]	Asignar el bit RLO al bit de datos de instancia cuya dirección exacta se encuentra en la palabra doble de datos DBD2.
L IB [DID 4]	Cargar en el ACU 1 el byte de entrada cuya dirección exacta se encuentra en la palabra doble de datos de instancia DID4.
OPN DB [LW2]	Abrir el bloque de datos cuyo número se encuentra en la palabra de datos local LW2.
O Q [LD 3] u O Q [juan]	Ejecutar una combinación O con el bit de salida cuya dirección exacta se encuentra en la palabra doble de datos locales 3 o en la dirección denominada "Juan" en la tabla de símbolos, como referencia a LD3.

Direccionamiento Indirecto por Memoria con un Número (III)

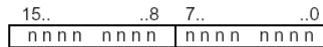
El direccionamiento indirecto con un número puede considerarse como otro tipo de direccionamiento el *Direccionamiento Indexado*.
 Es necesaria la utilización de un valor numérico de 16 bits (WORD) que debe de encontrarse en una de las siguientes áreas:

- Marcas (MW)
- Bloques de Datos Globales (DBW)
- Datos Locales (LW)
- Bloques de Datos Instancia (DIW)

Los Punteros de Palabra contienen números y se utilizan para direccionar elementos como:

- *Temporizadores (T)*
- *Funciones (FC's)*
- *Contadores (C)*
- *Bloques de Función (FB's)*
- *Bloques de Datos (DB's)*

Formato del Puntero de Palabra

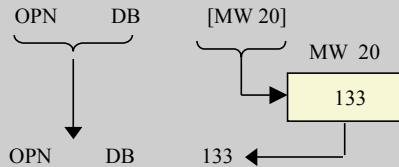


Bits 0 a 15 (nnnn nnnn nnnn nnnn): número (margen de 0 a 65.535) de un temporizador (T), contador (C), bloque de datos (DB), función (FC) o bloque de función (FB)

Direccionamiento Indirecto por Memoria (IV)

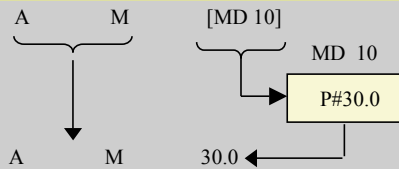
Direccionamiento Indirecto por Memoria con Número

```
L 133           //cargar puntero en ACU1
T MW 20        //salvar en palabra de memo.
OPN DB[MW20]  //abrir DB global
.....
SP T[MW 20]   //arrancar temporizador
```



Direccionamiento Indirecto por Memoria con Puntero a Área

```
L P#30.0       //cargar puntero en ACU1
T MD 10        //salvar en palabra doble
L MW[MD10]    //asignar dirección digital
.....
A M[MD 10]    //asignar dirección binaria
```



Direccionamiento Indirecto por Registro (I)

Direccionamiento Indirecto por Registro e Intraárea

- La dirección del área interna está definida en uno de los dos registros de direcciones (AR1 ó AR2).
- El contenido del registro de direcciones es un puntero a área interna
- Con el direccionamiento indirecto por registro, se especifica un *offset* que se suma al registro de direcciones
- La suma se realiza cuando la operación es ejecutada sin modificar el contenido del registro de direcciones
- El offset tiene el formato de un puntero a área
- En direccionamiento indirecto de direcciones digitales, el offset debe tener como dirección bit “0”
- El máximo valor es: P#8191.7

Ejemplo	Descripción
A I [AR1, P#4.3]	Ejecutar una operación lógica Y con el bit de entrada cuya dirección exacta se calcula sumando 4 bytes y 3 bits al contenido del registro de direcciones AR1.
= DIX [AR2, P#0.0]	Asignar el bit RLO al bit de datos de instancia cuya dirección exacta se encuentra en el registro de direcciones AR2.
L IB [AR1, P#100.0]	Cargar en el ACU 1 el byte de entrada cuya dirección exacta se calcula sumando 100 bytes al contenido del registro de direcciones AR1.
T LD [AR2, P#56.0]	Transferir el contenido del ACU 1 a la palabra doble de datos local es LD, cuya dirección exacta se calcula sumando 56 bytes al contenido del registro de direcciones AR2. Por lo que respecta al direccionamiento de datos locales, tenga en cuenta la indicación que aparece más abajo.

Direccionamiento Indirecto por Registro (II)

Direccionamiento Indirecto por Registro e Interárea

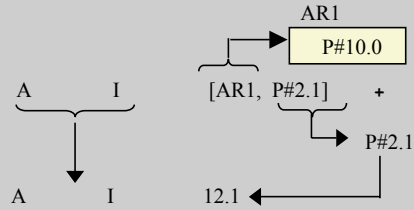
- La dirección general está definida en uno de los dos registros de direcciones (AR1 ó AR2).
- El contenido del registro de direcciones es un puntero en general
- Con el direccionamiento general, se escribe el área de direcciones junto con el puntero a área dentro del registro de direcciones
- Con direccionamiento indirecto sólo se especifica un identificador para la dirección:
No especificar un bit, “B” para byte, “W” para palabra y “D” para doble palabra
- Se especifica con un offset con dirección bit

Ejemplo	Descripción
A [AR1, P#4.3]	Ejecutar una operación lógica Y con el bit cuya dirección exacta se calcula sumando 4 bytes + 3 bits al contenido del registro de direcciones AR 1. El área de memoria del bit está indicada en los bits 24, 25 y 26 del registro de direcciones AR 1.
= [AR2, P#0.0]	Asignar el bit RLO al bit cuya dirección exacta se encuentra en el registro de direcciones AR 2. El área de memoria del bit está indicada en los bits 24, 25 y 26 del registro de direcciones AR 2.
L B [AR1, P#100.0]	Cargar en el ACU 1 el byte cuya dirección exacta se calcula sumando 100 bytes al contenido del registro de direcciones AR 1. El área de memoria del byte está indicada en los bits 24, 25 y 26 del registro de direcciones AR 1.
T D [AR2, P#56.0]	Transferir el contenido del ACU 1 a la palabra doble cuya dirección exacta se calcula sumando 56 bytes al contenido del registro de direcciones AR 2. El área de memoria de la palabra doble está indicada en los bits 24, 25 y 26 del registro de direcciones AR 2.

Direccionamiento Indirecto por Registro (III)

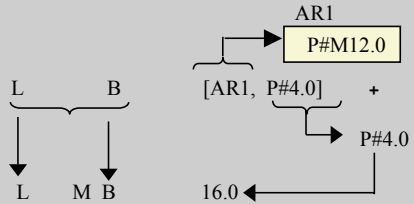
Direccionamiento Indirecto por Registro e Intraárea

```
LAR1 P#10.0 //cargar puntero dir. reg. 1
.....
T MW[AR1,P#4.0] //asignar dirección digital
.....
A I[AR1,P#2.1] //asignar dirección binaria
```



Direccionamiento Indirecto por Registro e Interárea

```
LAR1 P#M12.0 //cargar puntero dir. reg. 1
.....
L B[AR1,P#4.0] //asignar dirección digital
.....
= [AR1, P#0.7] // asignar dirección binaria
```



Operaciones Relacionadas con los Registros

Operaciones relacionadas con AR1 y AR2

Operación	Operando	Función
LAR1		Cargar contenido de ACU1 en registro de direcciones AR1.
LAR1	AR2	Cargar contenido del AR2 en registro de direcciones AR1.
LAR1	<Dirección>	Carga el contenido del área direccionada en el registro de direcciones AR1.
LAR1	P#{área}, byte{,bit}	Cargar el contenido de la dirección en registro de direcciones AR1.
LAR2		Cargar contenido de ACU1 en registro de direcciones AR2.
LAR2	<Dirección>	Carga el contenido del área direccionada en el registro de direcciones AR2.
LAR2	P#{área}, byte{,bit}	Cargar el contenido de la dirección en registro de direcciones AR2.
TAR		Intercambia el contenido de AR1 y AR2.
TAR1		Transfiere el contenido del registro de direcciones 1 AR1 al ACU1.
TAR1	AR2	Transfiere contenido del AR1 en registro de direcciones AR2.
TAR1	<Dirección>	Transfiere el contenido del registro de direcciones 1 AR1 al destino direccionado.
TAR2		Transfiere el contenido del registro de direcciones 2 AR2 al ACU1.
TAR2	<Dirección>	Transfiere el contenido del registro de direcciones 2 AR2 al destino direccionado.
+AR1		Suma el contenido de la palabra baja del ACU 1 al contenido del registro AR1.
+AR1	P#Byte.Bit	Suma una constante puntero al contenido del registro de direcciones AR1.
+AR2		Suma el contenido de la palabra baja del ACU 1 al contenido del registro AR2.
+AR2	P#Byte.Bit	Suma una constante puntero al contenido del registro de direcciones AR2.

Ejemplos: Direccionamiento Indirecto (I)

Ejemplo 1:

```

L      11
T      MW 60
OPN    DB [MW60]  →  OPN DB 11
    
```

Ejemplo 2:

```

L      P#24.0
T      MD 50
L      IW [MD50] → L IW 24
    
```

↑ ↑ ↑ Dirección
 Tipo de Acceso
 Zona de Acceso

Ejemplos: Direccionamiento Indirecto (II)

Ejemplo 3:

AR1	31 00000000 00000RRR	23 00000000 00000BBB	15 00000000 BBBBBBBBB	7 01010000 BBBBBXXX
-----	----------------------------	----------------------------	-----------------------------	---------------------------

LAR1 P#10.0 // Preselección
 L MW [AR1, P#200.0] → +200 → L MW 210

Tipo de Dato Zona de Acceso Registro Dirección Constante de Desfase Instrucción Ejecutada

Ejemplo 4:

AR1	31 10000001 10000RRR	23 00000000 00000BBB	15 00000011 BBBBBBBBB	7 00100000 BBBBBXXX
-----	----------------------------	----------------------------	-----------------------------	---------------------------

LAR1 P#I 100.0 // Preselección
 L B [AR1, P#110.0] → +210 → L IB 210

Zona de Acceso Registro Dirección Constante de Desfase Instrucción Ejecutada

Ejemplos de Direccinamiento (III)

Ejemplo 5: Memoria con Nmero

```
L 5
T DBW100
A T[DBW100]
```

Ejemplo 6: Memoria con Puntero

```
L P#0.0
T MD 100
A I[MD 100]
L MD[MD 100]
```

Ejemplo 7: Registro Intrarea

```
LAR1 P#0.0
L 8
M01: T MW 20
A I[AR1, P#0.0]
= Q[AR1, P#4.0]
+AR1 P#0.1
L MW 20
LOOP M01
```

Ejemplo 8: Registro Interrea

```
LAR1 P#I0.0
LAR2 P#Q4.0
L 8
M02: T MW 20
A [AR1, P#0.0]
= [AR2, P#0.0]
+AR1 P#0.1
+AR2 P#0.1
L MW 20
LOOP M02
```

Ejemplos: Direccinamiento Indirecto (IV)

Ejemplo 9:

Comparacin STEP5 STEP7

STEP 7		STEP 5
L 11		L KF11
T MW 60		T MW 60
OPN DB[MW 60]	→	OPN DB 11
		(A DB 11) ← A DB 0

L P#24.0		L KF 24
T MD 50		T MW 50
L IW [MD50]	→	L IW 24
		B MW 50
		L IW 0