

# **Como se puede Trabajar con los PLC 16xx**

**(En bajo nivel)**

## **Capitulo 8**

### **Instrucciones Básicas.**

**LD : (4C)**

Carga el primer contacto de la línea o rama.

**LDN: (42)**

Carga el primer contacto de la línea o rama en forma negada

**AND: (41)**

Lógica "Y"

**ANDN: (44)**

Lógica "Y" negada

**OR: (4F)**

Lógica "O"

**ORN: (47)**

Lógica "O" negada

**= ( Copy ) (49)**

Copia la condición de la rama o línea.

**=NOT:(4E)**

Niega la condición de la rama o línea.

**=SET: (50)**

Coloca un "1" en el lugar de las acciones y permanece así independientemente que la condición de halla borrado

**=RST: (52)**

ídem a =SET pero coloca un "0" en el lugar de las acciones

**=JMP: (4A)**

Salta a una línea o rama especificada.

**IN: (45)**

Especifica una Entrada Digital

**OUT: (53)**

Especifica una Salida Digital

**MEN: (4D)**

Especifica un relé Interno

**TIM : (54)**

Lee el bit de estado del Timers

**CNT: (43)**

Lee el bit de estado del Contador

**ENDSUB : (464646)**

Fin de subrutina.

**END. (58).**

Fin de Programa

***Importante.***

Para comunicarse con el PLC, desde cualquier terminal de datos, se debe Setear el puerto serie al que esta conectado ( COM 1 a COM 4) y los parámetros de comunicación, 9600,N,8,1, y el terminal en modo ASCII. Todo esto es posible realizarlo si no se posee el software de programación y simulación bajo Windows que al adquirir el PLC se provee. Como software para enviar estos comandos puede ser cualquiera de los que comúnmente se comercializa por la firma Microsoft, dentro del los paquetes de Windows 95 / 98 (Hyperterminal), o cualquier otro que funcione bajo DOS, por ejemplo Procomm ó PC-plus, etc.

### ***Leer un Programa:***

Solamente en modo **STOP**, al enviar desde cualquier programa de comunicaciones la **letra (R)** se recibirá el contenido del programa existente en la memoria del PLC.

### ***Ejemplo:***

Supongamos que en la memoria del PLC, esta cargado un programa simple, que en lista de instrucciones seria de la siguiente manera:

```
LD in 1
= Out 1
End
```

Es decir si la **In 1** esta actuada, la **Out 1**, debe accionar.

Entonces, si oprimimos la **letra R**, siempre y cuando el PLC este detenido, recibiremos lo siguiente:

**4C450149530146464658.**

### ***Escribir un Programa:***

Por ejemplo si deseamos realizar una **OR** entre la **entrada 1 ( In1)** y la **entrada 5 ( In5)**, y su resultado enviarlo a la **salida 3 ( Out3)** se debe proceder de la siguiente manera:

**W4C45014F450549530346464658.**

Debemos recordar que para esta instancia el PLC deberá estar detenido.

### ***Borrar el contenido de la Memoria del PLC:***

Cuando el PLC esta detenido al oprimir desde cualquier terminal de datos la **letra (D)**, **borraremos la totalidad** de la memoria **NVRAM** del PLC, es decir **datos y programa**

### ***Ejecutar un Programa:***

Para que el PLC comience a ejecutar el programa que tiene cargado en su memoria debemos oprimir el botón superior marcado como **RUN / STOP**, o mediante cualquier terminal de datos enviar la **letra (G)**.

**Detener la Ejecución:**

Para que el PLC deje de ejecutar el programa que tiene cargado en su memoria debemos oprimir el botón superior marcado como **RUN / STOP**, o mediante cualquier terminal de datos enviar la **letra (S)**.

**Leer Entradas, Salidas o Memorias**

Solamente en modo **RUN**, procediendo de igual forma que con los anteriores comandos, al ingresar desde una terminal de datos las letras **(i)** ó **(o)** ó **(m)** leeremos el estado de las entradas, salidas y memorias.

Por ejemplo si tenemos activas las **entradas 1, 3 y la 15**, y oprimo la **letra (i)** desde un terminal de datos, recibiremos el siguiente mensaje:

**05C0**

**Construcción de los mensajes:**

Entradas																											
8	7	6	5					4	3	2	1					16	15	14	13					12	11	10	9
Peso								Peso								Peso								Peso			
8	4	2	1					8	4	2	1					8	4	2	1					8	4	2	1
Ent. Activa								Ent. Activa								Ent. Activa								Ent. Activa			
0								5								C								0			

De igual forma se procederá con las salidas, si nos interesa saber el estado de estas, y suponiendo que tenemos activas las **salidas 4, 5, 9 y 16**, y oprimo la **letra (o)** desde un terminal de datos, recibiremos el siguiente mensaje:

**1A81**

Salidas																											
8	7	6	5					4	3	2	1					16	15	14	13					12	11	10	9
Peso								Peso								Peso								Peso			
8	4	2	1					8	4	2	1					8	4	2	1					8	4	2	1
Sal. Activa								Sal. Activa								Sal. Activa								Sal. Activa			
1								A								8								1			

Con las memorias se trabaja de forma similar, pero al enviar la **letra (m)** desde un terminal de datos, recibiremos la totalidad de ellas. Por ejemplo si tenemos activas las memorias **3, 4, 12, 21, 62 y la 64**, recibiremos el siguiente mensaje:

**0C081000000000A0**

### **Setear Entradas, Salidas o Memorias**

Solamente en modo **RUN**, si deseamos setear las **entradas 4, 5, 7, 8, 9, 12, 15 y 16** debemos proceder de la siguiente forma y tener en cuenta que:

Entradas Primer Paquete															
8	7	6	5					4	3	2	1				
Peso								Peso							
8	4	2	1					8	4	2	1				
Entrada a Setear								Entrada a Setear							
D								8							

Se procederá a enviar por el terminal de datos el siguiente comando para setear las entradas marcadas del primer paquete.:

**I00D8**

Para trabajar con el segundo paquete de entradas **in 9 a in 16**, procederemos de la siguiente forma:

Entradas Segundo Paquete								
16	15	14	13		12	11	10	9
Peso					Peso			
8	4	2	1		8	4	2	1
Entrada a Setear					Entrada a Setear			
C					9			

Se procederá a enviar por el terminal de datos el siguiente comando para setear las entradas marcadas del segundo paquete.:

**I01C9**

Para el seteo de las Memorias del PLC se procederá de igual forma que para las entradas y salidas, tenga en cuenta también que debemos realizarlo por paquetes de ocho memorias por paquete y empezaremos a contar del paquete cero al siete, por consiguiente tendremos para trabajar sesenta y cuatro memoria, entre comunes y especiales, recuerde que el tercer paquete corresponde a memoria especiales que más adelante explicaremos.

También tenga en cuenta que todas las memorias son retentivas, es decir que ante un corte de energía mantienen su valor.

Por ejemplo si deseo setear las memoria del primer paquete, **MEM 3 y MEM 7** procederemos de la siguiente manera:

Memorias Primer Paquete (M00)								
8	7	6	5		4	3	2	1
Peso					Peso			
8	4	2	1		8	4	2	1
Memoria a Setear					Memoria a Setear			
4					4			

Se procederá a enviar por el terminal de datos el siguiente comando para setear las memorias marcadas del primer paquete.:

**M0044**

Ahora realizaremos otro ejemplo de seteos de memorias del octavo paquete es decir de **MEM 56 a MEM 64** y setearemos **MEM 57, MEM 58, MEM 63 y MEM 64**:

Memorias Octavo Paquete (M07)								
64	63	62	61		60	59	58	57
Peso					Peso			
8	4	2	1		8	4	2	1
Memoria a Setear					Memoria a Setear			
C					3			

Se procederá a enviar por el terminal de datos el siguiente comando para setear las memorias marcadas del octavo paquete:

**M07C3**

### ***Leer Timers y Contadores:***

#### ***Lectura de valor de seteo:***

Para leer el valor de seteo ó predeterminación de cualquier temporizador ó contador se procederá de la siguiente manera, introduciremos el numero de timers ó contador, (**T01 a T16**) ó (**C01 a C16**):

Se procederá a enviar por el terminal de datos el siguiente comando para leer el valor de seteo, por ejemplo del **Timers 1**:

**T01**

Y el mensaje que recibiremos será el valor de predeterminación del **timers 1** expresado en hexadecimal, (**01 a FF**), si estaba seteado en 15 (bases de tiempo), nos enviara:

**0F**

De igual manera procederemos con los contadores, por ejemplo si deseamos saber el valor de predeterminación del contador cinco, se procederá a enviar por el terminal de datos el siguiente comando para leer el valor de seteo:

**C05**

Y el mensaje que recibiremos será el valor de predeterminación del **contador 5** expresado en hexadecimal, (**01 a FF**), si estaba seteado el valor 64 nos enviara:

**40**

***Lectura del valor actual:***

Para leer el valor actual de cualquier temporizador ó contador se procederá de la siguiente manera, introduciremos él numero de timers ó contador, (**T01 a T16**) ó (**C01 a C16**):

Se procederá a enviar por el terminal de datos el siguiente comando para leer el valor actual, por ejemplo del **Timers 15**:

**t0F**

Y el mensaje que recibiremos será el valor actual del **Timers 15** expresado en **hexadecimal**, (**01 a FF**), por ejemplo

**CA**

Y veremos que sé ira decrementando a medida que transcurre el tiempo y esta expresado en unidades de base de tiempo.

De igual manera procederemos con los contadores, por ejemplo si deseamos saber el valor actual del **Contador 10**, se procederá a enviar por el terminal de datos el siguiente comando para leer el valor actual:

**c0A**

Y veremos que sé ira decrementando a medida que el contador recibe pulsos en la entrada de cuenta. El valor recibido **estará expresado en hexadecimal** y comprendido entre los valores **00 y FF**. Por ejemplo.

**AA**

### **Setear Timers y Contadores:**

Para setear el valor de predeterminación de cualquier temporizador ó contador se procederá de la siguiente manera, introduciremos el numero de timers ó contador, ( **T01 a T16**) ó (**C01 a C16**) y luego el valor que deseamos setearle, expresado en **Hexadecimal**.

Por ejemplo si deseamos setearle al **Timer 4** un valor de cuenta de **20 base de tiempo**, se procederá a enviar por el terminal de datos el siguiente comando:

**T0414**

De igual forma se procederá con los contadores. Por ejemplo si deseamos setearle al **Contador 8** un valor de cuenta igual a **35**, se procederá a enviar por el terminal de datos el siguiente comando:

**C0823**

#### **Importante:**

**Cabe recordar que tanto los temporizadores, como los contadores son del tipo descendente, es decir al llegar a cero se produce su acción, también recordemos que los temporizadores T01 a T08, son retardo en la conexión y de T09 a T16, son retardo en la desconexión.**

## **Memorias especiales ( MEM 25 a MEM 32).**

### **MEM 25:**

Esta Memoria cambia de estado cada 300 ms. ( Ciclo ON 300 ms, ciclo OFF 300ms.) Ciclo de actividad **600 mseg.**

### **MEM 26:**

Esta Memoria cambia de estado cada 500 ms. ( Ciclo ON 500 ms, ciclo OFF 500 ms.) Ciclo de actividad **1 seg.**

### **MEM 27:**

Este PLC, puede trabajar como **placa de adquisición de señales discretas (I/O)**, para ello se procederá de la siguiente forma:

Pasar al PLC al modo **STOP**, luego se setea la **MEM 27**, Cortar la alimentación **PWR y Campo**, del PLC, esperar 10 seg. y volver a conectar. El sistema a partir de este momento funcionara en modo **STOP**, como una placa de entrada salida,

Para volver a funcionar como PLC, se reseteará la **MEM 27**, y repetir la operación descrita anteriormente.

Esta Placa permitirá desarrollar prácticas de programación con distintos lenguajes como ser **Pascal, Basic, C++, Visual Basic, Visual C, Assembler**, etc.

**Dydec, provee las librerías adicionales par trabajar con estos lenguajes.**

Asimismo permite desarrollar automatizaciones de 24 a 32 I/O brindando la posibilidad de conectar hasta **31 unidades en red**, mediante el protocolo de comunicaciones **MODBUS** y el módulo **adaptador RS- 232 a RS-485**.

Puede ser comandada por medio de Computadoras Personales PC , por medio del puerto serie. Posee entradas de tipo P Optoacopladas y sus salidas dependiendo del modelo, son cuatro del tipo Transistor (P) y cuatro del tipo relé, también Optoacopladas., en el modelo PLCem168 y para el modelo PLCem1616 son sus entradas de tipo P y sus 16 salidas transistorizadas, también del tipo P.

**MEM 28:**

Al setear la **MEM 28**, el PLC opera solamente con el **Protocolo MODBUS** en modo **RUN**, perdiendo la comunicación con el **Software VisualPLC**, y obedecerá los comandos a través de un **SCADA**, para dicho **Protocolo**. Por defecto tomará el **Alias 01**.

**MEM 29:**

**No usar, para funcionamiento normal del PLC, debe ser reseteada, ( reservada para uso exclusivo de Dydec).**

**MEM 30:**

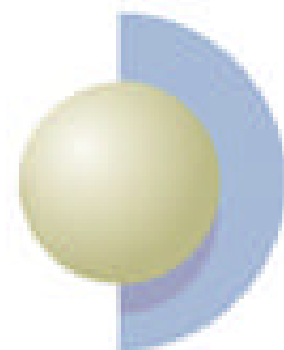
Esta Memoria cambia de estado cada 30 seg. ( **Ciclo ON 30 seg., Ciclo OFF 30 seg.**) Ciclo de actividad **1 minuto**.

**MEM 31:**

Esta Memoria cambia de estado cada 30 minutos. ( **Ciclo ON 30 minutos, Ciclo OFF 30 minutos.**) Ciclo de actividad **1 hora**

**MEM 32:**

**No usar, para funcionamiento normal del PLC, debe ser reseteada, ( reservada para uso exclusivo de Dydec).**



**Notas:**

**Notas:**

**Notas:**