

# ***Slicetex Ladder Designer Studio***

## **NOTA DE APLICACIÓN**

### **AN027**

### **Control de Display LCD**

*Autor: Ing. Boris Estudiez*



|                    |             |
|--------------------|-------------|
| Modelos Aplicables | AX, CX y DX |
|--------------------|-------------|

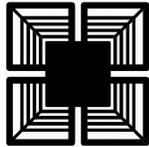
## **1 Descripción General**

---

*La presente nota de aplicación explica cómo utilizar el display LCD de texto incorporado en algunos modelos PLC.*

*El display LCD le permite imprimir en su pantalla mensajes de texto, valores de datos, etc. Para mayor información sobre las capacidades y características del display, consulte la hoja de datos del dispositivo.*

*Se proveen componentes en lenguaje Ladder y funciones en lenguaje Pawn para controlar el display LCD desde el PLC.*



## **2 Lecturas Recomendadas**

---

Antes de leer este documento, recomendamos que se familiarice con el software StxLadder y el PLC adquirido. Sugerimos leer los siguientes documentos:

1. Manual de Usuario del software StxLadder.
2. Manual de Programación Pawn del PLC (si utiliza lenguaje Pawn)
3. Hoja de datos técnicos del PLC.

Mas documentación puede encontrar en la página del producto: [www.slicetex.com](http://www.slicetex.com).

Para consultas y soporte, ponemos a disposición un foro de discusión en: [www.slicetex.com/foro](http://www.slicetex.com/foro)

En el foro puede leer preguntas de otros usuarios y realizar también sus propias consultas.

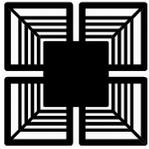
### **2.1 Ejemplos**

En nuestro sitio web, busque la pagina de la nota de aplicación AN027, allí podrá encontrar ejemplos completos para utilizar en el PLC.

## **3 Requerimientos**

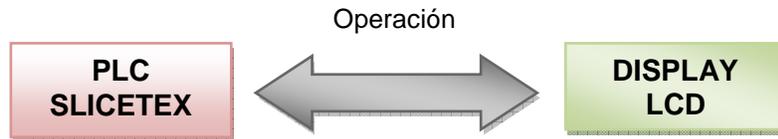
---

Para esta nota de aplicación, debe tener instalado en su computadora el entorno de Programación **StxLadder** (Slicetex Ladder) y utilizar un firmware actualizado con soporte para display LCD en el PLC.



## 4 Teoría de Funcionamiento

Algunos modelos de PLC de Slicetex Electronics incorporan un display LCD de texto donde es posible imprimir mensajes y valores de datos.



**Fig. 1: Operación entre PLC y display LCD.**

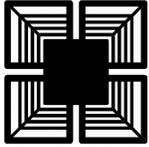
El tamaño del display LCD configurado en el dispositivo por defecto es de 16 caracteres por 2 líneas (16x2). Pero puede utilizar otros formatos de display (por ejemplo, 20x2, 16x4, 20x4, etc), siempre y cuando sea compatible con el controlador de display soportado por el dispositivo (remítase a la hoja de datos). En caso de tener un display de diferente tamaño al por defecto 16x2, deberá configurar el nuevo tamaño en su script con la función `LcdSize()`.

También verifique que su dispositivo no tiene desactivado el display LCD, el cual puede ser activado o desactivado a través del programa configuración **BoardConfig** o **StxLadder**.

Lea la nota de aplicación **AN002** si usa **BoardConfig** o el documento **STXLADDER-PC** si usa **StxLadder** para configurar el dispositivo.



**Fig. 2: Opciones para display LCD en BoardConfig o StxLadder**



## **5 Componentes para Lenguaje Ladder**

---

En esta sección explicaremos a modo general como utilizar el display LCD con el lenguaje Ladder. Puede bajar ejemplos completos de esta nota aplicación en nuestro sitio Web.

Para utilizar el display en lenguaje Pawn ir pagina 15.

### **5.1 Componentes**

Primero describiremos los componentes disponibles (al momento de escribir esta nota de aplicación, verifique última versión de StxLadder para componentes nuevos) que puede utilizar con el display LCD.

Recomendamos leer la descripción de los componentes primero, pero si quiere pasar directamente al ejemplo ir a pagina 11.

#### **5.1.1 Lcd Clear**



Agrega un componente que permite limpiar el texto de un display LCD.

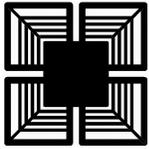
Este componente debe llamarse solo cuando sea necesario limpiar el display. Si se llama repetidamente a una frecuencia muy alta, puede generar un efecto visual llamado flickering (parpadeo) en el display LCD, que puede ser antiestético a fines prácticos.

El componente se ejecuta cuando el valor del flujo de corriente en el puerto de entrada “EN” es 1.

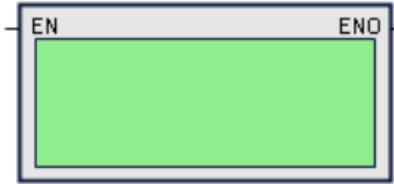
El valor del puerto de salida “ENO” es una copia del valor recibido en el puerto “EN”.

#### **Notas:**

- Este componente puede insumir hasta 2 mS de tiempo al ejecutarse debido a características internas del display LCD. Por lo que es un componente de ejecución relativamente lenta y debe llamarse solo cuando es necesario.



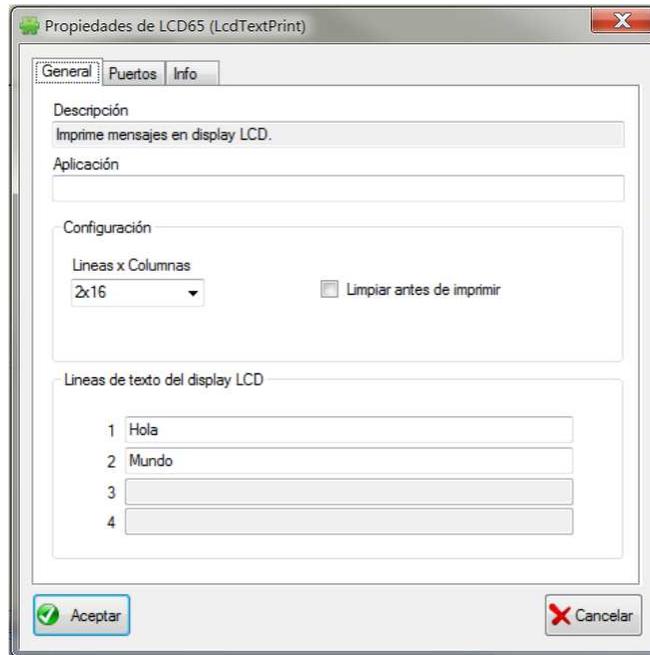
### 5.1.2 Lcd Text Print



Agrega un componente que permite imprimir mensajes de texto en un display LCD.

El componente se ejecuta cuando el valor del flujo de corriente en el puerto de entrada "EN" es 1.

El valor del puerto de salida "ENO" es una copia del valor recibido en el puerto "EN".



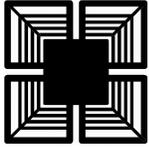
Ventana de propiedades del componente

#### **Funcionamiento:**

El texto a imprimir en cada línea del display se especifica en las propiedades del componente.

Si el mensaje excede la longitud del display, los caracteres se escriben en la línea siguiente disponible o al comienzo del display.

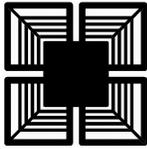
Para un correcto uso del componente, debe especificar el tamaño del display LCD (Líneas x Columnas) en las propiedades del mismo.



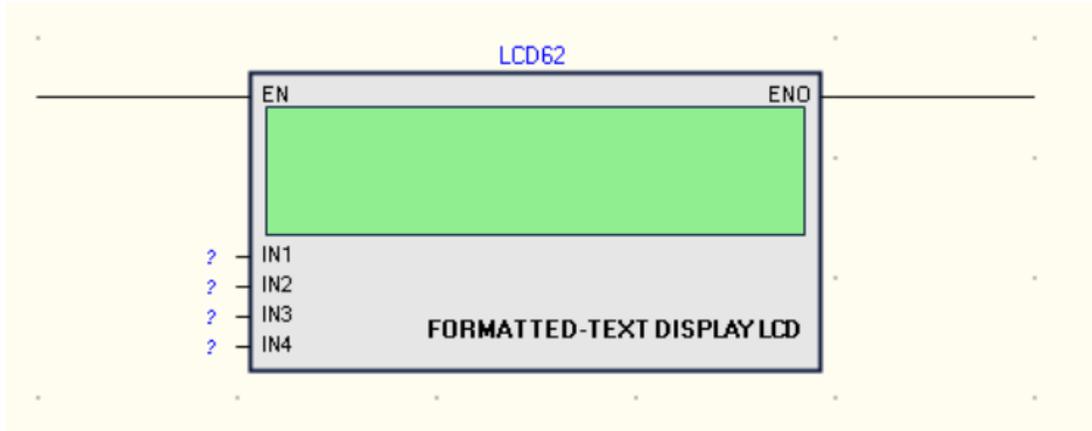
Es posible limpiar el LCD antes de escribir, pero esta opción puede retardar la ejecución del código 2 mS. La opción "***Limpiar antes de escribir***" puede utilizarse para tal fin.

Consejos:

- Para imprimir texto con formato, es decir mensajes y valores de variables combinados, utilice el componente "***LcdTextPrintf***". El tiempo de ejecución de dicho componente es superior.
- Puede limpiar el LCD imprimiendo espacios " " en todas las líneas del display. Este método es más eficiente que utilizar la opción "***Limpiar antes de imprimir***".



### 5.1.3 Lcd Formatted-Text Print



Agrega un componente que permite imprimir mensajes de texto con formato en un display LCD. La impresión con formato permite mezclar texto con valores de variables en la impresión de mensajes.

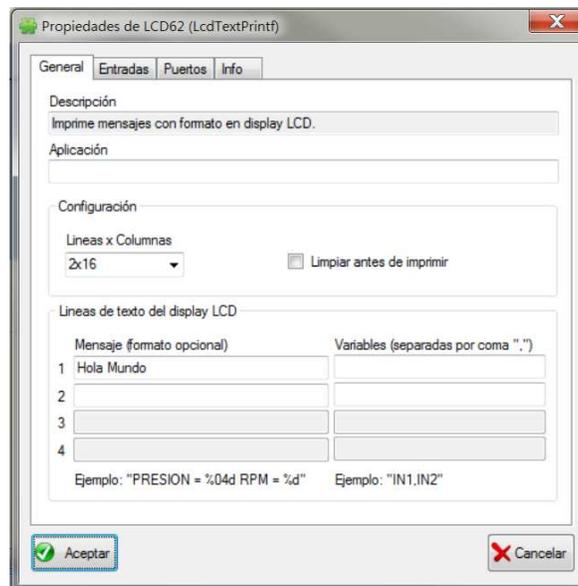
Si busca imprimir el valor de una variable, este es el componente indicado.

El texto a imprimir en cada línea del display se especifica en las propiedades del componente.

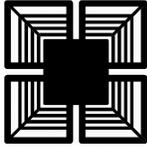
Si el mensaje excede la longitud del display, los caracteres se escriben en la línea siguiente disponible o al comienzo del display.

Para un correcto uso del componente, debe especificar el tamaño del display LCD (Líneas x Columnas) en las propiedades del mismo.

Es posible limpiar el LCD antes de escribir, pero esta opción puede retardar la ejecución del código 2 mS. La opción "**Limpiar antes de imprimir**" puede utilizarse para tal fin.



Ventana de propiedades del componente



**Entradas:**

El componente se ejecuta cuando el valor del flujo de corriente en el puerto de entrada "EN" es 1.

- Existen 4 entradas más, cuyo uso es opcional y permiten especificar las variables a leer e imprimir en la pantalla del LCD.
- Los puertos de entrada son "IN1", "IN2", "IN3" e "IN4". Dentro de las propiedades del componente, puede especificar las variables apuntadas por dichos puertos.

No es necesario utilizar todos los puertos al mismo tiempo, ni seguir un orden especial.

El tipo de variable o constante que puede seleccionar es amplio, puede ser **Int32**, **Float**, **String**, etc.

Lea la sección de formato para imprimir valores de variables.

**Salidas:**

El valor del puerto de salida "ENO" es una copia del valor recibido en el puerto "EN".

**Formato:**

Para imprimir mensajes con formato, es necesario especificar "**códigos de formato**" en el mensaje.

Los **códigos de formato** indican cómo debe interpretarse e imprimirse una variable.

Los **códigos de formato** están precedidos por un símbolo "%". Por ejemplo "%d" indica que en ese lugar del texto, debe imprimirse una variable entera.

Cada "**código de formato**" lee la variable apuntada por algún puerto de entrada. Los puertos que deben leerse se especifican en las propiedades del componente, en una caja de texto adyacente al mensaje a imprimir.

Ejemplo, para imprimir los RPM de un motor en el display LCD, cuyo valor se especifica en una variable entera apuntada por el puerto "IN1", hacemos:

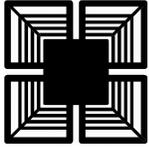
"RPM = %d" → "IN1"

En el display, se reemplazará el **código de formato** %d por el valor de la variable apuntada por el puerto "IN1".

Si deseamos imprimir el valor de RPM y PRESION, podemos hacer:

"RPM = %d PRESION = %d" → "IN1,IN2"

Notemos, como separamos con comas los puertos **IN1** e **IN2**.



Si tenemos un valor de tensión y deseamos imprimirlo como punto flotante, podemos hacer:

“Tension: %.3f [V]” → “IN1”

El resultado será el siguiente:

```
Tension: 11.345 [V]
```

Note como fue especificada la precisión, mostrando solo 3 dígitos decimales.

**Códigos de formatos disponibles:**

| Nombre | Tipo     | Descripción  |
|--------|----------|--|
| %d     | Entero   | Imprime un entero con signo.                             |
| %u     | Entero   | Imprime un entero sin signo.                             |
| %x     | Entero   | Imprime un entero con notación hexadecimal.              |
| %X     | Entero   | Imprime un entero con notación hexadecimal en mayúscula. |
| %f     | Flotante | Imprime un número decimal o punto flotante.              |
| %s     | Cadena   | Imprime una cadena de caracteres.                        |
| %c     | Caracter | Imprime un solo carácter.                                |

Un Entero es un valor tipo **Int32**.

Un Flotante es un valor tipo **Float**

Una Cadena es un valor tipo **String**.

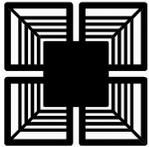
**Alineación y Relleno:**

Se antepone un numero entre el “%” y el código. El número especifica la cantidad de espacios que deberán ser rellenos antes de imprimir la variable. Si el dígito es negativo “-“, la alineación será hacia la izquierda.

Si antes del dígito de alineación, se antepone el numero 0, en vez de rellenar con espacios, la alineación será rellena con ceros.

**Precisión:**

En un número flotante es posible especificar la cantidad de dígitos decimales a imprimir (por defecto es dos). Por ejemplo, “%.000f” imprime un numero decimal con tres dígitos luego del punto solamente.



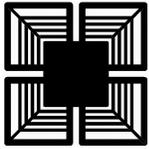
Ejemplos de Formatos:

| Ejemplo  | Argumento | Salida en Display (cada columna es un carácter) |   |   |   |   |   |   |  |
|----------|-----------|---|---|---|---|---|---|---|--|
| %d       | 3         | 3   |   |   |   |   |   |   |  |
| %4d      | 3         |   |   |   | 3 |   |   |   |  |
| %04d     | 3         | 0   | 0 | 0 | 3 |   |   |   |  |
| %-4d     | 3         | 3   |   |   |   |   |   |   |  |
| %04d     | -3        | -   | 0 | 0 | 3 |   |   |   |  |
| %4d      | -3        |   |   | - | 3 |   |   |   |  |
| %03.000f | 5.12345   | 0   | 5 | . | 1 | 2 | 3 |   |  |
| %f       | 5.12345   | 5   | . | 1 | 2 |   |   |   |  |
| %.4f     | 5.12345   | 5   | . | 1 | 2 | 3 | 4 | 5 |  |
| %.02f    | 3.1       | 3   | . | 1 | 0 |   |   |   |  |
| %x       | 255       | f   | f |   |   |   |   |   |  |
| %X       | 255       | F   | F |   |   |   |   |   |  |
| %s       | "HOLA"    | H   | O | L | A |   |   |   |  |
| %c       | 66        | B   |   |   |   |   |   |   |  |
| %c       | 'B'       | B   |   |   |   |   |   |   |  |
| %s       | 0         | (   | n | u | l | l | ) |   |  |

En las tablas, el término "Argumento" se refiere al valor del puerto (IN1, IN2, etc).

Notas / Consejos:

- No se pueden imprimir líneas de texto vacías o que solo imprimen espacios. En dicho caso, no será generado código para la impresión de la línea vacía. Note que algunas líneas pueden estar vacías, mientras que otras no, lo cual es permitido.
- Para imprimir texto sin formato, es decir mensajes únicamente, utilice el componente "**LcdTextPrint**". El tiempo de ejecución de dicho componente es inferior.
- Si usted es un programador en lenguaje C, el uso de este componente le resultara similar a la función "**printf()**" de dicho lenguaje.
- Algunos PLC no disponen de un display LCD incorporado, pero puede utilizar los componentes NetHMI.

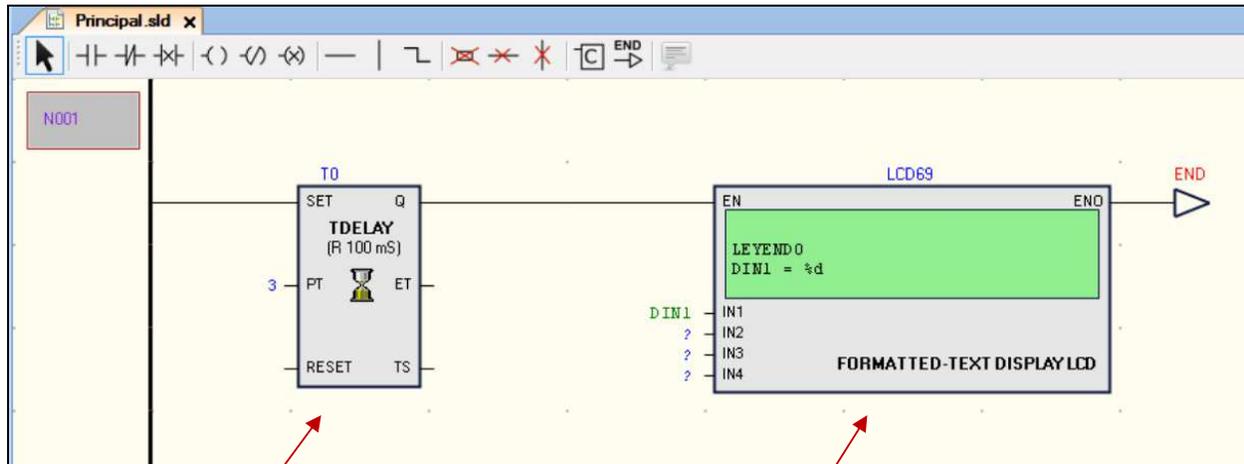


## 5.2 Ejemplo en Ladder para Display LCD

Realizaremos un ejemplo muy simple que nos permitirá familiarizarnos de forma práctica con el display LCD.

*Nota: Puede descargar los ejemplos de esta nota de aplicación desde la sección de archivos.*

El programa leerá cada 300 mS la entrada digital **DIN1** e imprimirá su valor en el display LCD, el programa Ladder completo se muestra en la siguiente figura:



Timer tipo TDELAY  
(Componente **TimerTDELAY**)

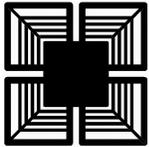
Display LCD  
(Componente **LcdTextPrintf**)

El código de arriba es muy simple:

En el diagrama "Principal" del proyecto Ladder se agrega en la network N001 dos componentes en serie:

- **TimerTDELAY**: Cada 300 mS el valor de la salida Q se pone en "1" por un solo ciclo del SCAN-CYCLE. La entrada PT configura el tiempo del timer.
- **LcdTextPrintf**: Imprime el valor de DIN1 en el display cuando la salida del timer **Q** es igual a 1, es decir cada 300 mS. El valor de DIN1 se lee desde el puerto IN1 del display.

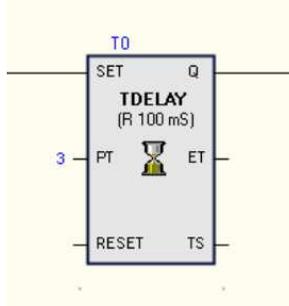
La salida **Q** del **TimerTDELAY** se mantiene en "1" por solo un SCAN-CYCLE, de esta forma el componente **LcdTextPrintf** se ejecuta **una sola vez**, de lo contrario, si no estuviera el timer se ejecutaría en cada SCAN-CYCLE produciendo un parpadeo en el display por la velocidad de refresco elevada.



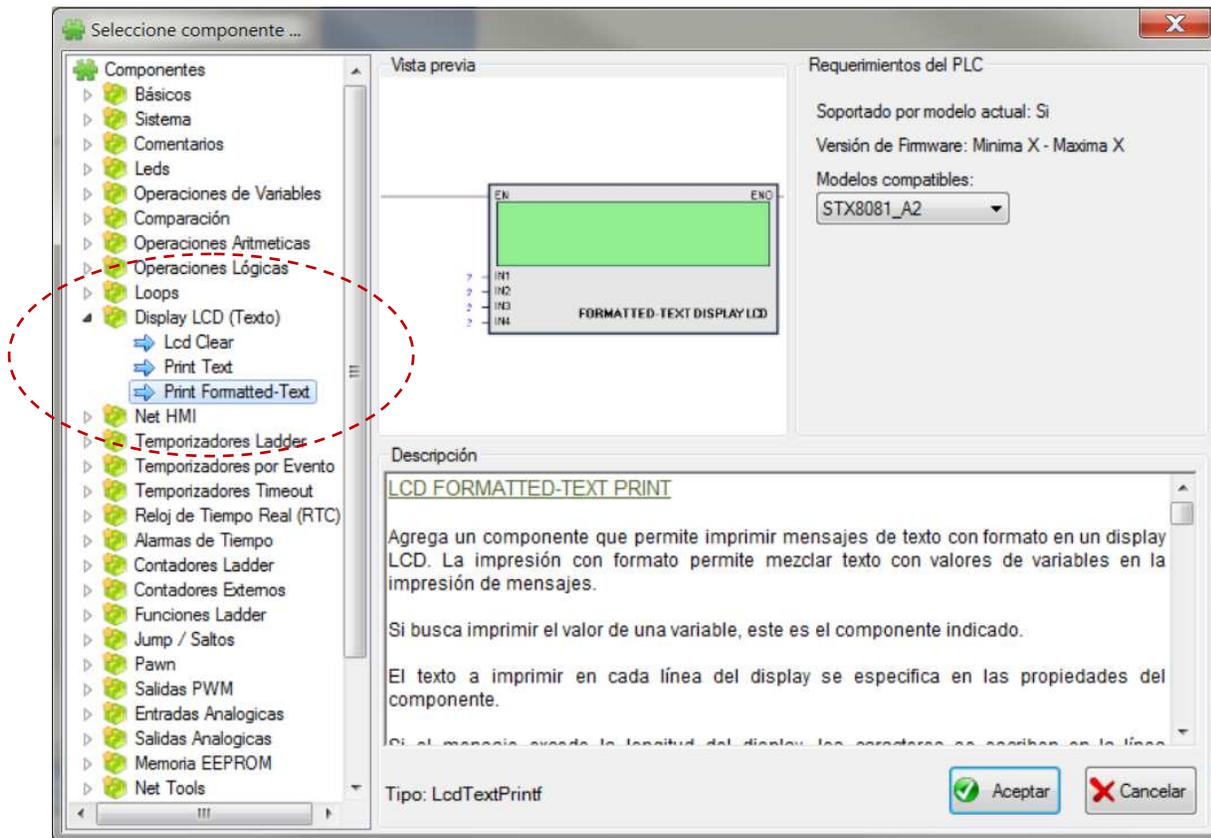
Para seleccionar un componente del display LCD, haga click en el icono siguiente ubicado en la barra de herramientas del diagrama:

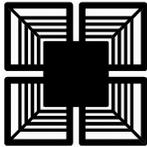


Busque dentro del grupo **“Temporizadores Ladder”** e inserte el componente **“TDELAY”** en la network **N001**. Doble click en el componente para acceder a las propiedades del mismo e ingrese el valor 3 en el puerto **“PT”** del componente. Esto producirá un retardo de 300 mS.



Luego en la celda contigua al timer TDELAY, agregue el componente para el display. Para ello en la ventana con los componentes disponibles dirijase al grupo **“Display LCD (Texto)”** como muestra la siguiente figura:





Seleccione el componente **"LcdTextPrintf"** bajo el nombre **"Print Formatted-Text"** y luego click en **"Aceptar"**.

El componente insertado se mostrará en el diagrama.

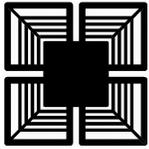
Doble click en las propiedades del componente **LcdTextPrintf** y se abrirá la siguiente pantalla:



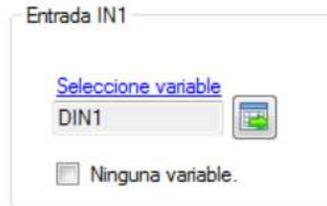
Seleccione **"Limpiar antes de imprimir"** para que se borre todo el contenido antes de imprimir y luego ingrese en los cuadros de texto lo siguiente:

|   | Mensaje (formato opcional) | Variables (separadas por coma ",") |
|---|----------------------------|------------------------------------|
| 1 | LEYENDO                    |                                    |
| 2 | DIN1 = %d                  | IN1                                |
| 3 |                            |                                    |
| 4 |                            |                                    |

Como se aprecia en la figura superior, se imprimirá en la primer línea el mensaje **"LEYENDO"**. En la segunda línea **"DIN1 = %d"**, donde **"%d"** será reemplazado por el valor de la variable leída en el puerto **IN1** como se especifico en la columna **Variables** (recuerde que %d es un código de formato, vea la descripción del componente en pagina 7).

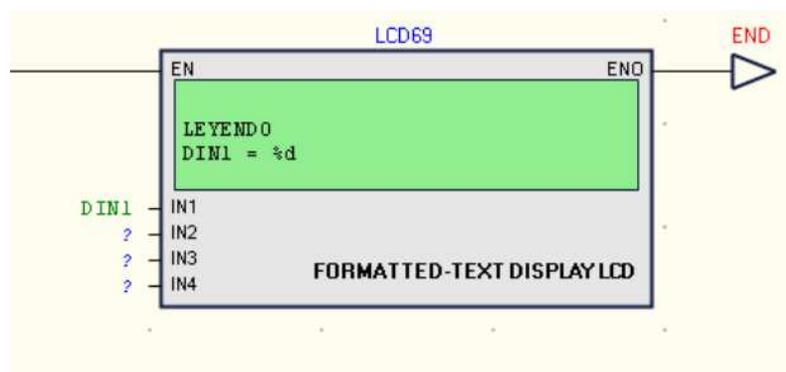


El próximo paso es especificar la variable del puerto **IN1**, para ello dirijase a la pestaña “Entradas” del componente. Habilite la entrada “**IN1**” u seleccione la variable **DIN1**.



En este caso se selecciona la variable **DIN1** que es de tipo booleana y solo tiene dos posibles valores “1” o “0”. Pero puede seleccionar una variable tipo Int32 si lo desea para mostrar números enteros diferentes.

Finalmente haga click en “Aceptar” y el componente insertado se mostrará como la siguiente figura:

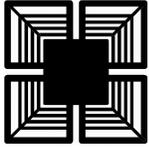


Note como “**DIN1**” se especifica en el puerto **IN1**. Además como el código de formato **%d** luego será reemplazado por el valor de **DIN1** cuando el PLC ejecute el programa.

Compile y cargue el programa en el PLC con apretando la tecla “F8”.

Pruebe el programa polarizando la entrada digital **DIN1** en el PLC. En el display se debería mostrar su estado.

¿ Simple verdad ?



## 6 Funciones Lenguaje Pawn

En esta sección explicaremos a modo general como utilizar el display LCD con el lenguaje Pawn. Puede bajar ejemplos completos de esta nota aplicación en nuestro sitio Web.

Para utilizar el display en lenguaje Ladder ir pagina 4.

### 6.1 Ejemplo en Pawn para Display LCD

Para comenzar, realizaremos un ejemplo muy simple que nos permitirá familiarizarnos de forma práctica con el display LCD.

El programa tiene el siguiente objetivo:

1. Leer estado de la entrada digital DIN1.
2. Almacenar la lectura en una variable.
3. Imprimir el estado en el display LCD.
4. Repetir el proceso (punto 1) cada 300 ms.

#### 6.1.1 Código

El proyecto de este programa se provee junto con los archivos de esta nota de aplicación con el nombre "PawnLcdFirtst". El código del script es el siguiente:

```
// Función principal.
PlcMain()
{
    // Crear variable para almacenar estado.
    new Din1Stat = 0

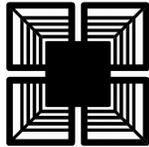
    // Limpiar LCD e imprimir mensaje de Bienvenida.
    LcdClear()
    LcdPrintf(0, 0, "LEYENDO...")

    // Loop principal.
    while(true)
    {
        // Leer entrada digital DIN1 y almacenar en variable Din1Stat
        Din1Stat = DinValue(DIN1)

        // Imprimir valor de variable "Din1Stat"
        LcdPrintf(0, 1, "DIN1 = %04d", Din1Stat)

        // Retardar ejecución del programa por 300 ms.
        DelayMS(300)

        // Conmutar led "DEBUG".
        LedToggle()
    }
}
```



En la función principal, primero creamos una variable llamada "Din1Stat" donde se almacenará la lectura de la entrada digital DIN1:

```
new Din1Stat = 0
```

Luego limpiamos o borramos la pantalla del LCD con la función:

```
LcdClear()
```

A continuación, imprimimos en la columna 0 y línea 0 del display LCD el mensaje: "LEYENDO...".

```
LcdPrintf(0, 0, "LEYENDO...")
```

Dentro del loop principal (observar que es infinito, nunca retorna), leemos la entrada DIN1 e imprimimos su valor en el LCD, cada 300 mS:

```
// Leer entrada digital DIN1 y almacenar en variable Din1Stat
Din1Stat = DinValue(DIN1)

// Imprimir valor de variable "Din1Stat"
LcdPrintf(0, 1, "DIN1 = %04d", Din1Stat)

// Retardar ejecución del programa por 300 ms.
DelayMS(300)

// Conmutar led "DEBUG".
LedToggle()
```

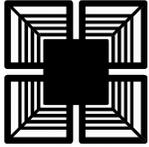
La función DinValue(DIN1) realiza una lectura del estado de la entrada DIN1, retornando 1 o 0, según el estado de la entrada. El valor devuelto se almacena en la variable "Din1Stat".

La variable "Din1Stat" es impresa en la columna 0, línea 1, del display LCD con la función:

```
LcdPrintf(0, 1, "DIN1 = %04d", Din1Stat)
```

El "%04d" le dice a la función LcdPrintf() que el próximo argumento es un entero con signo y deben imprimirse solo 4 dígitos, anteponiendo ceros a la izquierda. Si solo escribimos "%d", se imprime el número sin restricciones.

La función DelayMS(300) genera un retardo en el programa y la función LedToggle() solo conmuta el led debug del dispositivo. Este led es de propósito general en modo PLC y puede ser utilizado para cualquier fin.



### 6.1.2 Funcionamiento

Luego de compilar y cargar el script al PLC (como se explicó en el manual de programación Pawn), reseteamos el PLC y la pantalla LCD debería mostrar el siguiente texto:

```
LEYENDO...  
DIN1 = XXXX
```

Donde el valor "XXXX" puede ser 0000 o 0001, dependiendo del valor de la entrada digital **DIN1**.

Finalmente, si su dispositivo contiene entradas puede utilizar el siguiente código para leer tensión de la entrada analógica **VIN1**:

```
PlcMain()  
{  
    // Crear variable para almacenar voltaje.  
    new Float: Voltaje = 0  
  
    // Especificar para VIN1 el rango de 0 a 10V.  
    VinSetRange(1, VIN_RANGE_10V)  
  
    // Loop principal.  
    while(true)  
    {  
        // Leer entrada analógica VIN1 y almacenar en variable "Voltaje".  
        Voltaje = Float: VinReadVolt(1)  
  
        // Imprimir valor de variable "Voltaje"  
        LcdPrintf(0, 1, "VIN1=%02.2f [V]", Voltaje)  
  
        // Pausar por 300 ms y Conmutar led DEBUG.  
        DelayMS(300)  
        LedToggle()  
    }  
}
```

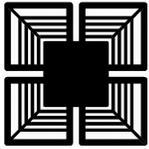
El display LCD mostrará si coloca 2.5V de tensión en la entrada VIN1:

```
LEYENDO...  
VIN1 = 02.50 [V]
```

La línea siguiente:

```
LcdPrintf(0, 1, "VIN1=%02.2f [V]", Voltaje)
```

Imprime el **Voltaje** en la línea 1 del display LCD (segunda línea). A la función **LcdPrintf()** le decimos que el primer argumento "Voltaje" luego del texto es un "Float" con el símbolo "%02.2f". El 02 indica que solo dos dígitos de parte entera deben imprimirse anteponiendo ceros y utilizar solo 2 decimales luego del punto. También puede no especificarse formato alguno, con "%f" solamente.



## 6.2 Funciones

LcdBackOn(): Activa el "backlight" (o luz trasera) del display.

| Argumentos | Tipo | Descripción                     |
|------------|------|---------------------------------|
| -          | -    |                                 |
| Retorno    | Tipo | Descripción                     |
| 0          | S    | Operación exitosa.              |
| -1         | S    | Error, display LCD desactivado. |

Ejemplo:

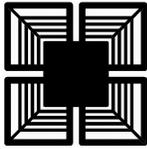
```
// Activar backlight.  
LcdBackOn()
```

LcdBackOff(): Apaga el "backlight" (o luz trasera) del display.

| Argumentos | Tipo | Descripción                     |
|------------|------|---------------------------------|
| -          | -    |                                 |
| Retorno    | Tipo | Descripción                     |
| 0          | S    | Operación exitosa.              |
| -1         | S    | Error, display LCD desactivado. |

Ejemplo:

```
// Apagar backlight.  
LcdBackOff()
```



LcdClear(): Limpia la pantalla del display (borra contenido).

| Argumentos | Tipo | Descripción  |
|------------|------|--|
| -          | -    |  |
| Retorno    | Tipo | Descripción  |
| 0          | S    | Operación exitosa.   |
| -1         | S    | Error, display LCD desactivado.  |
| Notas      |      | Descripción  |
| 1          |      | Esta función puede tardar entre 1.53 mS y 2 mS en retornar. Durante este tiempo, no se procesaran eventos. |
| 2          |      | Una función muy útil puede ser LcdClrLine(x, y) para borrar parcialmente una línea del display.            |

Ejemplo:

```
// Limpiar pantalla.  
LcdClear()
```

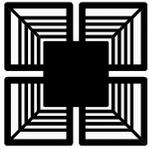
LcdSize(Lines, Columns): Especifica el tamaño del display que se utilizará en el PLC.

| Argumentos | Tipo | Descripción   |
|------------|------|---|
| Lines      | E    | Numero de líneas del display LCD.   |
| Columns    | E    | Numero de columnas del display LCD.   |
| Retorno    | Tipo | Descripción   |
| 0          | S    | Operación exitosa.  |
| -1         | S    | Error, display LCD desactivado.   |
| Notas      |      | Descripción   |
| 1          |      | Esta función debe ser llamada si su display es de tamaño distinto a 2 líneas por 16 columnas. |

Ejemplo:

Si tiene un display de 4 líneas por 20 caracteres puede utilizar:

```
// Especificar tamaño del display.  
LcdSize(4, 20)
```



LcdPutChar(Ch): Imprime un carácter y mueve el cursor a la próxima posición valida del display.

| Argumentos | Tipo | Descripción                       |
|------------|------|-----------------------------------|
| Ch         | E    | Carácter, valor de 8-bits, ASCII. |
| Retorno    | Tipo | Descripción                       |
| 0          | S    | Operación exitosa.                |
| -1         | S    | Error, display LCD desactivado.   |

Ejemplo:

Imprimir la palabra "HOLA" en pantalla, carácter, por carácter, hacemos:

```
// Imprimir carácter H.  
LcdPutChar('H')  
  
// Imprimir carácter O.  
LcdPutChar('O')  
  
// Imprimir carácter L.  
LcdPutChar('L')  
  
// Imprimir carácter A.  
LcdPutChar('A')
```

Notar, que el cursor avanza hacia la próxima posición valida del display (derecha) a medida que imprimimos un carácter.

LcdWriteCh(Ch): Imprime un carácter en el display. No mueve el cursor.

| Argumentos | Tipo | Descripción                       |
|------------|------|-----------------------------------|
| Ch         | E    | Carácter, valor de 8-bits, ASCII. |
| Retorno    | Tipo | Descripción                       |
| 0          | S    | Operación exitosa.                |
| -1         | S    | Error, display LCD desactivado.   |

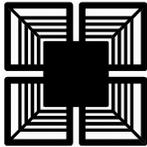
Ejemplo:

Si queremos imprimir la palabra "H" en pantalla, hacemos:

```
// Imprimir carácter H.  
LcdWriteCh('H')
```

Notar que si llamamos nuevamente a la función, se sobrescribirá el carácter impreso anteriormente, debido a que el cursor no se mueve.

```
// Imprimir carácter O.  
LcdWriteCh('O')
```



LcdMoveCursor(x, y): Mueve el cursor a la columna 'x' y línea 'y'.

| Argumentos | Tipo | Descripción  |
|------------|------|--|
| x          | E    | Columna.   |
| y          | E    | Línea.   |
| Retorno    | Tipo | Descripción  |
| 0          | S    | Operación exitosa.   |
| -1         | S    | Error, display LCD desactivado.  |
| Notas      |      | Descripción  |
| 1          |      | Si los valores exceden el tamaño del display, serán establecidos automáticamente a la primera columna o línea o ambas del display LCD. |

Ejemplo:

Imprimir la letra "B" en la columna 5 y línea 1 de la pantalla:

```
// Mover cursor a línea 1, columna 5.
LcdMoveCursor(5, 1)

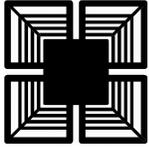
// Imprimir carácter B.
LcdPutChar('B')
```

LcdPrint(x, y, string[]): Imprime una cadena en el display, en posición x:y.

| Argumentos | Tipo | Descripción  |
|------------|------|--|
| X          | E    | Columna.   |
| Y          | E    | Línea.   |
| string     | E    | Cadena a imprimir.   |
| Retorno    | Tipo | Descripción  |
| 0          | S    | Operación exitosa.   |
| -1         | S    | Error, display LCD desactivado.  |
| Notas      |      | Descripción  |
| 1          |      | Si los valores exceden el tamaño del display, serán establecidos automáticamente a la primera columna o línea o ambas del display LCD. |
| 2          |      | Si los valores x e y, son iguales a '-1', la cadena es impresa en la posición actual del cursor en pantalla.                           |
| 3          |      | Esta función es más rápida que LcdPrintf() para imprimir solo cadenas.   |

Ejemplo:

```
// Imprimir "hola" en columna 1, línea 0.
LcdPrint(1, 0, "hola")
```



Si queremos imprimir " mundo" a continuación de la palabra anterior, hacemos:

```
// Imprimir " mundo".
LcdPrint(-1, -1, " mundo")
```

Notar, que al establecer x=-1 e y=-1, la función utilizó como coordenadas la posición actual del cursor, el cual se encontraba justo después del último carácter impreso de la palabra "hola". Esto puede ser muy útil para concatenar cadenas.

También es posible imprimir una cadena almacenada en una variable, por ejemplo:

```
// Variable para almacenar una cadena.
new Mensaje = "Hola Mundo"

// Imprimir variable Mensaje
LcdPrint(0, 0, Mensaje)
```

La variable "Mensaje" almacena una cadena en la memoria RAM. La cual puede ser modificada dentro del script. Si nuestra cadena no va a ser modificada, es decir del tipo constante, podemos almacenarla en la memoria ROM de la siguiente forma:

```
#define MENSAJE "Hola Mundo"

LcdPrint(0, 0, Mensaje)
```

Donde MENSAJE es una definición constante. La misma debe ser declarada al comienzo de script, antes de PlcMain(). Lea el manual de programación del lenguaje Pawn.

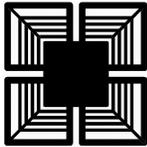
LcdClrLine(x, y): Borra caracteres desde la posición 'x:y' hasta el final de la línea.

| Argumentos | Tipo | Descripción                     |
|------------|------|---------------------------------|
| x          | E    | Columna.                        |
| y          | E    | Línea.                          |
| Retorno    | Tipo | Descripción                     |
| 0          | S    | Operación exitosa.              |
| -1         | S    | Error, display LCD desactivado. |

Ejemplo:

```
// Imprimimos "Hola Mundo"
LcdPrint(0,0, "Hola Mundo")

// Borramos solo la palabra "Mundo", que comienza en la columna 5, línea 0.
LcdClrLine(5,0)
```



| LcdPrintf(x, y, string[], ...): Imprime una cadena con formato en el display, en posición x:y. |      |  |
|--|------|--|
| Argumentos   | Tipo | Descripción  |
| X  | E    | Columna.   |
| Y  | E    | Línea.   |
| string   | E    | Cadena a imprimir, con o sin formato.  |
| ...  | E    | Lista de argumentos variables, de acuerdo a si se especifica formato.  |
| Retorno  | Tipo | Descripción  |
| 0  | S    | Operación exitosa.   |
| -1   | S    | Error, display LCD desactivado.  |
| Notas  |      | Descripción  |
| 1  |      | Si los valores exceden el tamaño del display, serán establecidos automáticamente a la primera columna o línea o ambas del display LCD. |
| 2  |      | Si los valores x e y, son iguales a '-1', la cadena es impresa en la posición actual del cursor en pantalla.                           |

La función **LcdPrintf()** es la más importante de las funciones para imprimir caracteres en el display LCD, ya que permite imprimir cadenas con formato de forma simple. Su utilidad es enorme. Por ello, haremos énfasis en explicar su funcionamiento.

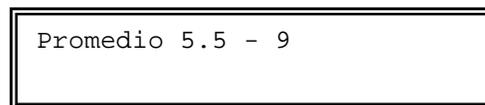
Su implementación es una versión simplificada de la clásica función **printf()** del lenguaje C, por lo tanto no incluye todas sus características.

Los primeros dos argumentos de la función seleccionan la coordenada en X e Y donde se imprimirá en el display la cadena. El tercer argumento, es la cadena con formato que será explicada más adelante. El último argumento, denominado "...", significa que puede incluir un número de argumentos variable. Con argumentos variables nos referimos a que el número de argumentos no está definido. De acuerdo al formato de la cadena a imprimir, los argumentos son opcionales y pueden o no pueden existir.

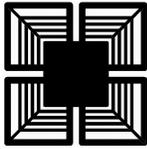
Un ejemplo básico de uso es:

```
LcdPrintf(0, 0, "Promedio %f - %d", 5.5, 9)
```

Imprimirá en pantalla:



Donde los códigos de formato, precedidos por un "%", son: "%f" y "%d". Indican que los argumentos que deben pasarse en la lista de argumentos variables, son un número flotante (Float) y otro entero con signo, en ese orden. También pueden pasarse los valores como variables.



Códigos Formato:

| Nombre | Tipo     | Descripción  |
|--------|----------|--|
| %d     | Entero   | Imprime un entero con signo.                             |
| %u     | Entero   | Imprime un entero sin signo.                             |
| %x     | Entero   | Imprime un entero con notación hexadecimal.              |
| %X     | Entero   | Imprime un entero con notación hexadecimal en mayúscula. |
| %f     | Flotante | Imprime un número decimal o punto flotante.              |
| %s     | Cadena   | Imprime una cadena de caracteres.                        |
| %c     | Caracter | Imprime un solo carácter.                                |

También es posible especificar alineación, relleno y precisión (en un numero punto flotante).

Alineación y Relleno:

Se antepone un numero entre el “%” y el código. El número especifica la cantidad de espacios que deberán ser rellenos antes de imprimir la variable. Si el dígito es negativo “-“, la alineación será hacia la izquierda.

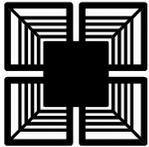
Si antes del dígito de alineación, se antepone el numero 0, en vez de relleno con espacios, la alineación será rellena con ceros.

Precisión:

En un número flotante es posible especificar la cantidad de dígitos decimales a imprimir (por defecto es dos). Por ejemplo, “%.000f” imprime un numero decimal con tres dígitos luego del punto solamente.

Ejemplos de formatos:

| Ejemplo  | Argumento | Salida en Display (cada columna es un carácter) |   |   |   |   |   |   |  |
|----------|-----------|---|---|---|---|---|---|---|--|
| %d       | 3         | 3   |   |   |   |   |   |   |  |
| %4d      | 3         |   |   |   | 3 |   |   |   |  |
| %04d     | 3         | 0   | 0 | 0 | 3 |   |   |   |  |
| %-4d     | 3         | 3   |   |   |   |   |   |   |  |
| %04d     | -3        | -   | 0 | 0 | 3 |   |   |   |  |
| %4d      | -3        |   |   | - | 3 |   |   |   |  |
| %03.000f | 5.12345   | 0   | 5 | . | 1 | 2 | 3 |   |  |
| %f       | 5.12345   | 5   | . | 1 | 2 |   |   |   |  |
| %.4f     | 5.12345   | 5   | . | 1 | 2 | 3 | 4 | 5 |  |
| %.02f    | 3.1       | 3   | . | 1 | 0 |   |   |   |  |

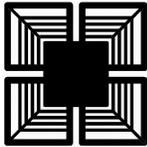


| Ejemplo | Argumento | Salida en Display (cada columna es un carácter) |   |   |   |   |   |  |  |
|---------|-----------|---|---|---|---|---|---|--|--|
| %x      | 255       | f   | f |   |   |   |   |  |  |
| %X      | 255       | F   | F |   |   |   |   |  |  |
| %s      | "HOLA"    | H   | O | L | A |   |   |  |  |
| %c      | 66        | B   |   |   |   |   |   |  |  |
| %c      | 'B'       | B   |   |   |   |   |   |  |  |
| %s      | 0         | (   | n | u | l | l | ) |  |  |

Ejemplos prácticos:

Nota: Remítase a los archivos de la nota de aplicación para ejemplos completos.

|   |   |                           |
|---|---|---------------------------|
| <pre>// Imprimir cadena. LcdPrintf(0,0, "Nombre: %s", "Rodya")</pre>  | → | Nombre: Rodya             |
| <pre>// Imprimir numero. LcdPrintf(0,0, "Edad: %d", 28)</pre>   | → | Edad: 28                  |
| <pre>// Imprimir decimal. LcdPrintf(0, 0, "Tanque: %f %%", 60.37)</pre>   | → | Tanque: 60.37 %           |
| <pre>// Imprimir Variables new Nombre[] = "Rodya", Edad = 28  LcdPrintf(0,0, "Nombre: %s", Nombre) LcdPrintf(0,1, "Edad: %d", Edad)</pre> | → | Nombre: Rodya<br>Edad: 28 |
| <pre>// Imprimir con relleno. LcdPrintf(0, 0, "Cuenta: %05d", 112)</pre>  | → | Cuenta: 00112             |
| <pre>// Imprimir sin relleno. LcdPrintf(0, 0, "Cuenta: %5d", 112)</pre>   | → | Cuenta: 112               |
| <pre>// Limitar precisión. LcdPrintf(0, 0, "Tension: %.3f [V]", 11.34556)</pre>   | → | Tension: 11.345 [V]       |



LcdHome(): Mueve el cursor a la posición x=0, y=0.

| Argumentos | Tipo | Descripción   |
|------------|------|---|
| -          | -    |   |
| Retorno    | Tipo | Descripción   |
| 0          | S    | Operación exitosa.  |
| -1         | S    | Error, display LCD desactivado.                             |
| Notas      |      | Descripción   |
| 1          |      | Esta función puede tardar entre 1.53 mS y 2 mS en retornar. |

Ejemplo:

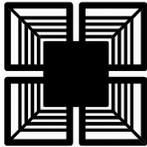
```
// Llevar el cursor a posición inicial.  
LcdHome()  
  
// Imprimir carácter B en la posición inicial.  
LcdPutChar('B')
```

LcdCursor(Op): Activa o desactiva el cursor la visualización del cursor en el display.

| Argumentos | Tipo | Descripción  |
|------------|------|--|
| Op         | E    | Si el valor pasado es ON, el cursor es mostrado. Si es OFF, el cursor se oculta. |
| Retorno    | Tipo | Descripción  |
| 0          | S    | Operación exitosa.   |
| -1         | S    | Error, display LCD desactivado.  |

Ejemplo:

```
// Mostrar cursor.  
LcdCursor(ON)  
  
// Ocultar cursor.  
LcdCursor(OFF)
```



| LcdCursorBlink(Op): Activa o desactiva el cursor el destello del cursor en el display. |      |   |
|--|------|---|
| Argumentos   | Tipo | Descripción   |
| Op   | E    | Si el valor pasado es ON, el cursor destella. Si es OFF, el cursor no destella. |
| Retorno  | Tipo | Descripción   |
| 0  | S    | Operación exitosa.  |
| -1   | S    | Error, display LCD desactivado.   |
| Notas  |      | Descripción   |
| 1  |      | Para utilizar esta función, primero active el cursor con LcdCursor(ON).         |

Ejemplo:

```
// Cursor destellante (Blink)
LcdCursorBlink(ON)

// Detener destello del cursor.
LcdCursorBlink(OFF)
```

| LcdDisplayOn(): Activa el display LCD. |      |                                 |
|--|------|---------------------------------|
| Argumentos                             | Tipo | Descripción                     |
| -                                      | -    |                                 |
| Retorno                                | Tipo | Descripción                     |
| 0                                      | S    | Operación exitosa.              |
| -1                                     | S    | Error, display LCD desactivado. |

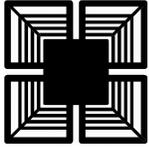
Ejemplo:

```
// Activa el display
LcdDisplayOn()
```

| LcdDisplayOff(): Desactiva el display LCD. |      |                                 |
|--|------|---------------------------------|
| Argumentos                                 | Tipo | Descripción                     |
| -  | -    |                                 |
| Retorno                                    | Tipo | Descripción                     |
| 0  | S    | Operación exitosa.              |
| -1   | S    | Error, display LCD desactivado. |

Ejemplo:

```
// Desactiva el display
LcdDisplayOff()
```

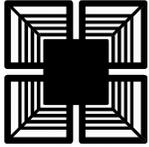


LcdPrintNum(x, y, num): Imprime un numero en el display, en posición x:y.

| Argumentos | Tipo | Descripción  |
|------------|------|--|
| x          | E    | Columna.   |
| y          | E    | Línea.   |
| num        | E    | Numero a imprimir.   |
| Retorno    | Tipo | Descripción  |
| 0          | S    | Operación exitosa.   |
| -1         | S    | Error, display LCD desactivado.  |
| Notas      |      | Descripción  |
| 1          |      | Si los valores exceden el tamaño del display, serán establecidos automáticamente a la primera columna o línea o ambas del display LCD. |
| 2          |      | Si los valores x e y son iguales a '-1', el numero se imprimirá en la posición actual del cursor en pantalla.                          |
| 3          |      | Esta función es más rápida que LcdPrintf() para imprimir solo números.   |

Ejemplo:

```
// Almacenar un numero en una variable.  
new Num = 67328  
  
// Imprimir la variable en la posición 0:0.  
LcdPrintNum(0, 0, Num)
```



LcdPrintHex(x, y, num): Imprime una numero en el display en formato hexadecimal, en posición x:y.

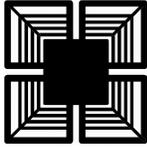
| Argumentos | Tipo | Descripción  |
|------------|------|--|
| x          | E    | Columna.   |
| y          | E    | Línea.   |
| num        | E    | Numero a imprimir.   |
| Retorno    | Tipo | Descripción  |
| 0          | S    | Operación exitosa.   |
| -1         | S    | Error, display LCD desactivado.  |
| Notas      |      | Descripción  |
| 1          |      | Si los valores exceden el tamaño del display, serán establecidos automáticamente a la primera columna o línea o ambas del display LCD. |
| 2          |      | Si los valores x e y son iguales a '-1', el numero se imprimirá en la posición actual del cursor en pantalla.                          |
| 3          |      | Esta función es más rápida que LcdPrintf() para imprimir solo números.   |

Ejemplo:

```
// Imprimir el número 255 en posición 0:0.  
LcdPrintHex(0, 0, 255)
```

En pantalla resulta:





LcdProgressBar(x, y, Progress, MaxProgress, Length): Dibuja una barra grafica de progreso en la posicion x:y. La misma muestra analógicamente el porcentaje entre "Progress" / "MaxProgress". El display tiene un tamaño de "Lengh" caracteres.

| Argumentos  | Tipo | Descripción  |
|-------------|------|--|
| x           | E    | Columna.   |
| y           | E    | Línea.   |
| Progress    | E    | Es el valor que la barra grafica debería mostrar.                          |
| MaxProgress | E    | Es el valor máximo que la barra grafica debería mostrar.                   |
| Length      | E    | Números de caracteres del display LCD que la barra grafica debería cubrir. |
| Retorno     | Tipo | Descripción  |
| 0           | S    | Operación exitosa.   |
| -1          | S    | Error, display LCD desactivado.  |

Ejemplo:

```
PlcMain()
{
    // Contador.
    new i = 0

    // Imprimir mensaje de bienvenida.
    LcdClear()
    LcdPrintf(0, 0, "Progress Bar:")

    // Ciclo principal.
    while(true)
    {
        // Imprimir una barra de progreso, con el valor "i" como indicador
        // y un tamaño de 12 caracteres de ancho.
        LcdProgressBar(0, 1, i, 255, 12)

        // Imprimir el valor "i" al lado de la barra.
        LcdPrintf(13, 1, "%03d", i)

        // Incrementar contador "i".
        i++
        // Limitar "i" al valor 255.
        if(i > 255)
        {
            i = 0
        }

        // Pausar programa por 300 mS.
        DelayMS(300)
    }
}
```



*Nota: El ejemplo está disponible en los archivos de la nota de aplicación.*

El ejemplo imprime una barra de progreso, la cual se va llenando a medida que la variable "i" se incrementa en 1. Cuando llega a 255 la barra esta completa, en este punto, a la variable "i" se le vuelve a asignar el valor inicial cero y el proceso se repite infinitamente.

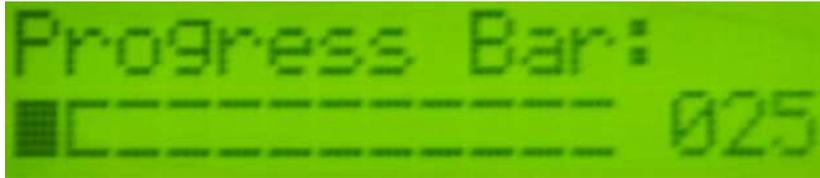
La línea:

```
LcdProgressBar(0, 1, i, 255, 12)
```

Imprime una barra de progreso grafica en posición 0:1 con un valor máximo de 255, un tamaño de 12 caracteres y un progreso igual al valor de la variable "i".

A continuación algunas fotografías del display LCD mostrando la barra de progreso:

Progreso en valor de i = 25:



Progreso cuando el valor de i = 255 (barra completa):





## **7 Abreviaciones y Términos Empleados**

---

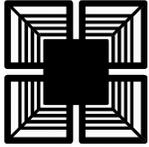
- **PLC:** Programmable Logic Controller (Controlador Lógico Programable).
- **Modo PLC:** Permite programar el dispositivo mediante un programa en Ladder o Pawn.
- **LCD:** Pantalla de Cristal Líquido. Muestra información visual.

## **8 Historial de Revisiones**

---

**Tabla 1: Historia de Revisiones del Documento**

| <b>Revisión</b>   | <b>Cambios</b> | <b>Descripción</b>              | <b>Estado</b> |
|-------------------|----------------|---------------------------------|---------------|
| 01<br>12/MAR/2014 | 1              | 1. Versión preliminar liberada. | Preliminar    |



## **9 Referencias**

---

Ninguna.

## **10 Información Legal**

---

### **10.1 Aviso de exención de responsabilidad**

**General:** La información de este documento se da en buena fe, y se considera precisa y confiable. Sin embargo, Slicetex Electronics no da ninguna representación ni garantía, expresa o implícita, en cuanto a la exactitud o integridad de dicha información y no tendrá ninguna responsabilidad por las consecuencias del uso de la información proporcionada.

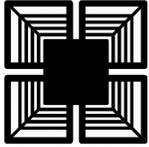
**El derecho a realizar cambios:** Slicetex Electronics se reserva el derecho de hacer cambios en la información publicada en este documento, incluyendo, especificaciones y descripciones de los productos, en cualquier momento y sin previo aviso. Este documento anula y sustituye toda la información proporcionada con anterioridad a la publicación de este documento.

**Idoneidad para el uso:** Los productos de Slicetex Electronics no están diseñados, autorizados o garantizados para su uso en aeronaves, área médica, entorno militar, entorno espacial o equipo de apoyo de vida, ni en las aplicaciones donde el fallo o mal funcionamiento de un producto de Slicetex Electronics pueda resultar en lesiones personales, muerte o daños materiales o ambientales graves. Slicetex Electronics no acepta ninguna responsabilidad por la inclusión y / o el uso de productos de Slicetex Electronics en tales equipos o aplicaciones (mencionados con anterioridad) y por lo tanto dicha inclusión y / o uso es exclusiva responsabilidad del cliente.

**Aplicaciones:** Las aplicaciones que aquí se describen o por cualquiera de estos productos son para fines ilustrativos. Slicetex Electronics no ofrece representación o garantía de que dichas aplicaciones serán adecuadas para el uso especificado, sin haber realizado más pruebas o modificaciones.

**Los valores límites o máximos:** Estrés por encima de uno o más valores límites (como se define en los valores absolutos máximos de la norma IEC 60134) puede causar daño permanente al dispositivo. Los valores límite son calificaciones de estrés solamente y el funcionamiento del dispositivo en esta o cualquier otra condición por encima de las indicadas en las secciones de Características de este documento, no está previsto ni garantizado. La exposición a los valores limitantes por períodos prolongados puede afectar la fiabilidad del dispositivo.

**Documento:** Prohibida la modificación de este documento en cualquier medio electrónico o impreso, sin autorización previa de Slicetex Electronics por escrito.



## ***11 Información de Contacto***

---

Para mayor información, visítenos en [www.slicetex.com](http://www.slicetex.com)

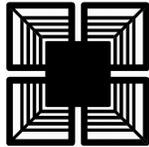
Para información, consultas y ventas, envíe un mail a: [info@slicetex.com](mailto:info@slicetex.com)

Para soporte técnico ingrese a nuestro foro en: [www.slicetex.com/foro](http://www.slicetex.com/foro)

Ing. Boris Estudiez

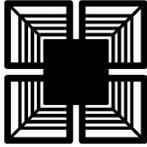
**Slicetex Electronics**  
Córdoba, Argentina

© Slicetex Electronics, todos los derechos reservados.



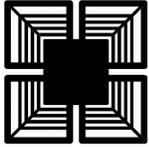
## **12 Contenido**

|                  |   |                  |
|------------------|---|------------------|
| <b><u>1</u></b>  | <b><u>DESCRIPCIÓN GENERAL.....</u></b>                | <b><u>1</u></b>  |
| <b><u>2</u></b>  | <b><u>LECTURAS RECOMENDADAS.....</u></b>              | <b><u>2</u></b>  |
| 2.1              | EJEMPLOS .....  | 2                |
| <b><u>3</u></b>  | <b><u>REQUERIMIENTOS .....</u></b>                    | <b><u>2</u></b>  |
| <b><u>4</u></b>  | <b><u>TEORÍA DE FUNCIONAMIENTO.....</u></b>           | <b><u>3</u></b>  |
| <b><u>5</u></b>  | <b><u>COMPONENTES PARA LENGUAJE LADDER .....</u></b>  | <b><u>4</u></b>  |
| 5.1              | COMPONENTES .....                                     | 4                |
| 5.1.1            | LCD CLEAR .....                                       | 4                |
| 5.1.2            | LCD TEXT PRINT .....                                  | 5                |
| 5.1.3            | LCD FORMATTED-TEXT PRINT .....                        | 7                |
| 5.2              | EJEMPLO EN LADDER PARA DISPLAY LCD .....              | 11               |
| <b><u>6</u></b>  | <b><u>FUNCIONES LENGUAJE PAWN.....</u></b>            | <b><u>15</u></b> |
| 6.1              | EJEMPLO EN PAWN PARA DISPLAY LCD .....                | 15               |
| 6.1.1            | CÓDIGO .....  | 15               |
| 6.1.2            | FUNCIONAMIENTO .....                                  | 17               |
| 6.2              | FUNCIONES.....  | 18               |
| <b><u>7</u></b>  | <b><u>ABREVIACIONES Y TÉRMINOS EMPLEADOS.....</u></b> | <b><u>32</u></b> |
| <b><u>8</u></b>  | <b><u>HISTORIAL DE REVISIONES.....</u></b>            | <b><u>32</u></b> |
| <b><u>9</u></b>  | <b><u>REFERENCIAS.....</u></b>                        | <b><u>33</u></b> |
| <b><u>10</u></b> | <b><u>INFORMACIÓN LEGAL .....</u></b>                 | <b><u>33</u></b> |
| 10.1             | AVISO DE EXENCIÓN DE RESPONSABILIDAD.....             | 33               |
| <b><u>11</u></b> | <b><u>INFORMACIÓN DE CONTACTO .....</u></b>           | <b><u>34</u></b> |
| <b><u>12</u></b> | <b><u>CONTENIDO .....</u></b>                         | <b><u>35</u></b> |



---

12.1 ÍNDICE DE TABLAS..... 37



### ***12.1 Índice de Tablas***

**Tabla 1: Historia de Revisiones del Documento ..... 32**

Copyright Slicetex Electronics 2015

[www.slicetex.com](http://www.slicetex.com)