



Slicetex Ladder Designer Studio

NOTA DE APLICACIÓN

AN028

Bus de Comunicaciones I2C

Autor: Ing. Boris Estudiez



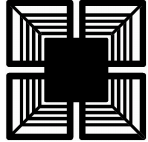
Modelos Aplicables	AX, CX y DX
--------------------	-------------

1 Descripción General

La presente nota de aplicación explica cómo utilizar la interface de comunicaciones I2C (Inter-Integrated Circuit) incorporado en el PLC (consulte hoja de datos por disponibilidad).

El puerto I2C permite conectar periféricos comerciales económicos al PLC, como conversores A/D, sensores de temperatura, sensores de humedad, conversores D/A, expansiones, etc. El puerto I2C es muy útil expandir las características del PLC con su propio hardware a un costo muy bajo.

Esta nota de aplicación explica el conexionado básico y las funciones disponibles en el PLC para programar el puerto I2C.



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.

Para consultas y soporte, ponemos a disposición un foro de discusión en: www.slicetex.com/foro

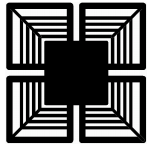
En el foro puede leer preguntas de otros usuarios y realizar también sus propias consultas.

2.1 Ejemplos

En nuestro sitio web, busque la pagina de la nota de aplicación AN028, 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 comunicación I2C en el PLC.



4 Teoría de Funcionamiento

La interfaz I2C le permite conectar dispositivos, chips o circuitos integrados que utilizan el bus I2C. Algunos ejemplos de estos dispositivos son: sensores de temperatura, humedad, convertidores A/D, convertidores D/A, relojes RTC, displays, etc. Es una interfaz de bajo nivel, lo que significa que interactúa directamente con sistemas electrónicos.

La ventaja del bus I2C radica en su simpleza, amplia disponibilidad comercial y bajo costo. La desventajas se centran en que hay que interpretar hojas de datos para programar o leer datos del sensor o dispositivos, la longitud del cable de conexión no debería exceder 1.5 metros para altas velocidades, posibilidad de ruidos en cableados largos o con múltiples periféricos conectados y posibilidad de daños eléctricos al manipularlo sin medidas de protección.

La mayoría de los modelos de PLC de Slicetex Electronics incorporan uno o más interfaces I2C. Consulte hoja de datos del PLC para disponibilidad.

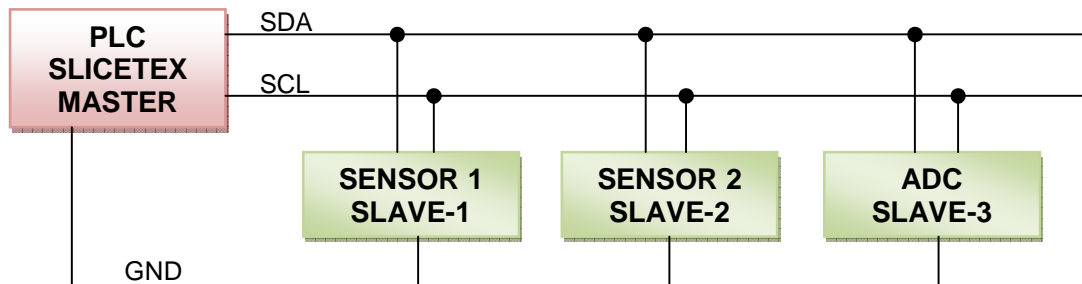


Fig. 1: Conexión típica I2C entre PLC y tres nodos esclavos.

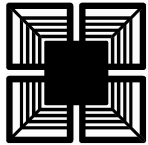
La Fig. 1 muestra el conexionado I2C típico entre el PLC y dispositivos I2C, llamados nodos. El PLC de Slicetex actúa como Master (Maestro) y controla a los Slaves (Esclavos) conectados como nodos, es decir al Sensor 1, Sensor 2 y al convertidor ADC.

Cada Slave tiene asociada una dirección I2C dentro del BUS, la misma es definida por el fabricante. Esta dirección es un entero de 7 bits (128 valores, de las cuales muchas están reservadas o son especiales). Se debe mencionar que dicha dirección debe ser única para cada Slave.

El Master es el único que inicia las comunicaciones a los Slaves para transmitir o recibir datos. El Master envía una transacción de escritura o lectura de datos a un Slave determinado utilizando su dirección I2C. Luego el Slave responde al Master confirmando la transacción.

Las líneas eléctricas de conexionado son SDA (Serial Data), SCL (Serial Clock) y GND (masa digital del PLC). Esos tres cables son en general los utilizados para conectar del PLC a sus dispositivos I2C.

La transmisión de datos se realiza por SDA, por SCL se transmiten pulsos de clock y GND es para compartir la referencia a tierra de voltaje entre circuitos (se puede aislar galvánicamente la interfaz con circuitos especiales).



5 Conexión al PLC

El PLC puede disponer de una o más interfaces I2C, llamadas I2C0, I2C1, I2C2, etc. Debe consultar la hoja de datos para disponibilidad.

Para conectar su dispositivo utilice como referencia el siguiente diagrama:

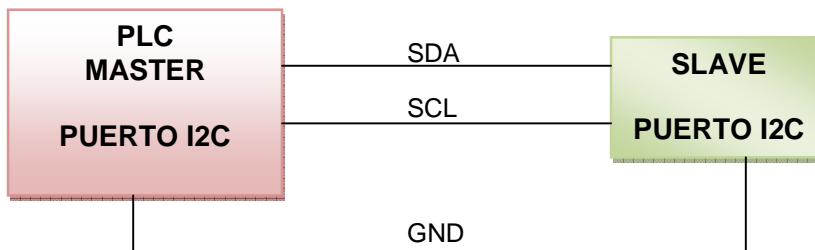


Fig. 2: Conexión I2C

En base a la Fig. 2, siga los siguientes pasos:

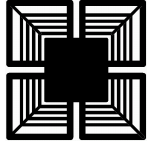
1. Consulte la hoja de datos del PLC e identifique la interfaz I2C a utilizar.
2. Identifique los terminales SDA y SCL de la interfaz I2C seleccionada. La mayoría de los PLC tienen una interfaz I2C en el puerto de expansión. Pero esto puede diferir en algunos modelos y número de interfaz I2C seleccionada.
3. Identifique el terminal GND o masa digital del PLC. Puede encontrarse en un puerto de expansión o bornera.
4. **Importante:** Desconecte energía eléctrica de todo el sistema (PLC y dispositivo I2C). Esto es muy importante, de otra manera puede quemar el puerto I2C del PLC. **Nunca conecte o desconecte cables I2C con el sistema energizado ni manipule los terminales con estática en la mano, de otra forma puede dañar la electrónica.**
5. Conecte mediante un cable los terminales SDA, SCL y GND del PLC a los terminales SDA, SCL, y GND del dispositivo I2C (que puede ser una placa, un chip, un sensor, etc).
6. Energice el sistema y en este punto ya puede empezar a utilizar la interfaz I2C del PLC.

A menos que la hoja de datos del PLC especifique lo contrario, los terminales SDA y SCL de la interfaz I2C del PLC, incorporan una resistencia de 2.2KOhms conectada a +5Vcc (pull-up). La tensión máxima de dichos terminales no debe superar los +6Vcc. Consulte con hoja de datos.

Si tiene dudas o necesita más información, le recomendamos ampliamente consultar antes en nuestro foro de soporte técnico sobre el I2C para evitar dañar su PLC:

www.slicetex.com/foro

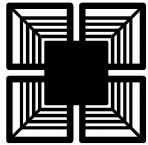
Nota: Consúltenos por venta de cables I2C a medida con conectores.



6 Componentes para Lenguaje Ladder

En esta sección explicaremos a modo general como utilizar la interfaz I2C con el lenguaje Ladder.

**Sección en elaboración, a completar.
Disculpe.**



7 Funciones Lenguaje Pawn

En esta sección explicaremos a modo general como utilizar la interfaz I2C con el lenguaje Pawn. Puede bajar ejemplos completos de esta nota aplicación en nuestro sitio Web.

Para utilizar la interfaz I2C en lenguaje Ladder ir pagina 5.

7.1 Consideraciones

En esta nota de aplicación se le llama I2Cx de forma genérica a las funciones para operar las interfaces I2C0, I2C1, I2C2, etc que pueden estar disponibles en el PLC. Donde la “x” representa el numero de interfaz 0, 1, 2, etc

Por ejemplo: nombramos la función **I2CxMasterSend()**, donde la “x” puede hacer referencia a **I2C0MasterSend()** para la interfaz I2C0. Si se desea utilizar la interfaz I2C1, entonces la función debe llamarse **I2C1MasterSend()**.

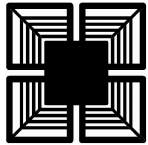
7.2 Tabla de Códigos I2C retornados

La siguiente tabla contiene los códigos I2C retornados por funciones al realizar una transacción I2C.

Nombre de constante definida	Valor	Descripción
I2C_LIB_SUCCESS	0	Operación exitosamente completada!.
I2C_LIB_MTR_ERR	-1	Error Transmisión/Recepción en modo Master.
I2C_LIB_TIMEOUT	-2	Error, timeout.
I2C_LIB_TO_BUSY	-3	Error, timeout y librería todavía ocupada.
I2C_LIB_BUSY	-4	Error, Librería ocupada.
I2C_LIB_FUNC_ERR	-5	Error interno de función.
I2C_LIB_MTR_W_ERR	-6	Error en transmisión Master.
I2C_LIB_MTR_R_ERR	-7	Error en recepción Master.
I2C_LIB_MTR_A_ERR	-8	Error, pérdida de arbitración en modo Master.
I2C_LIB_MW_STA_ERR	-9	Error de transmisión, no se pudo enviar START.
I2C_LIB_MW_NOACK_ERR	-10	Error, datos transmitidos, pero no se recibió ACK.
I2C_LIB_MW_NOADR_ERR	-11	Error, el esclavo no respondió un ACK de su dirección.
I2C_LIB_MR_STA_ERR	-12	Error en recepción de datos, no se pudo enviar START.
I2C_LIB_OP_TIMEOUT	-13	Error, hardware I2C timeout.

Tabla 1: Códigos I2C retornados

Como vemos en la Tabla 1, los códigos de error I2C son negativos. Mientras que para una operación exitosa el número 0 es devuelto.



7.3 Funciones I2C

A continuación se listan las funciones I2C disponibles para operar la interfaz.

7.3.1 Transmisión de Datos

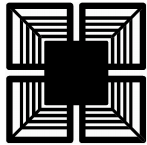
Las siguientes funciones transmiten o escriben datos del PLC al dispositivo I2C.

I2CxMasterSendByte(Addr, Byte): Envía un byte sobre el bus I2C número "x".		
Argumentos	Tipo	Descripción
Addr	E	Dirección I2C del dispositivo esclavo.
Byte	E	Byte de dato a enviar / escribir.
Retorno	Tipo	Descripción
0	S	Operación exitosa.
-1 a -20	S	Error, ver códigos retornados en Tabla 1 en página 6.
Notas		Descripción
1		La función genera las condiciones START, STOP y verificación de ACK requeridas por el bus automáticamente.

Ejemplo:

Enviar un byte con valor 55 al dispositivo I2C en dirección 0xD0 utilizando interfaz I2C0:

```
// Realizar transacción I2C para escribir byte.  
I2C0MasterSendByte(0xD0, 55)
```

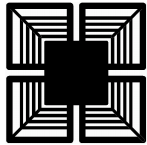


I2CxBusSend(Addr, Length, Data[]): Envía un array de bytes sobre el bus I2C número "x".		
Argumentos	Tipo	Descripción
Addr	E	Dirección I2C del dispositivo esclavo.
Length	E	Cantidad de bytes a enviar / escribir (máx. 81).
Data[]	E	Array con los bytes a enviar. Cada elemento del array es un byte.
Retorno	Tipo	Descripción
0	S	Operación exitosa.
-1 a -20	S	Error, ver códigos retornados en Tabla 1 en página 6.
Notas	Descripción	
1	La función genera las condiciones START, STOP y verificación de ACK requeridas por el bus automáticamente.	

Ejemplo:

Enviar un 8 bytes con valores al dispositivo I2C en dirección 0xD0 utilizando interfaz I2C0:

```
// Crear array con datos.  
new Data[8]  
  
Data[0] = 23  
Data[1] = 53  
Data[2] = 77  
Data[3] = 89  
Data[4] = 57  
Data[5] = 78  
Data[6] = 22  
Data[7] = 11  
  
// Enviar array de datos.  
I2C0MasterSend(0xD0, 8, Data)
```

I2C0xMasterSendProtocol(Addr, Header[], HeaderLength, Data[], DataLength, Trailer[], TrailerLength, ErrorCheck=1):		
Toma varios array de bytes y los concatena antes de enviar sobre el bus I2C número "x".		
Argumentos	Tipo	Descripción
Addr	E	Dirección I2C del dispositivo esclavo.
Header[]	E	Array con los bytes a enviar de header. Cada elemento del array es un byte.
HeaderLength	E	Cantidad de bytes a enviar / escribir de header (máx. 10).
Data[]	E	Array con los bytes a enviar de data. Cada elemento del array es un byte.
DataLength	E	Cantidad de bytes a enviar / escribir de data (máx. 81).
Trailer[]	E	Array con los bytes a enviar de trailer. Cada elemento del array es un byte.
TrailerLength	E	Cantidad de bytes a enviar / escribir de tráiler (máx. 10).
ErrorCheck[]	E	Comprueba errores si es 1. Si es 0, ignora error "Slave Address ACK" en el BUS.
Retorno	Tipo	Descripción
0	S	Operación exitosa.
-1 a -20	S	Error, ver códigos retornados en Tabla 1 en página 6.
Notas	Descripción	
1	La función genera las condiciones START, STOP y verificación de ACK requeridas por el bus automáticamente.	

Ejemplo:

Esta función es muy útil para combinar secuencias de valores antes de enviarla a un dispositivo I2C.

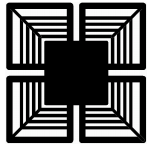
Supongamos querer enviar comandos a un dispositivo, de los cuales tenemos 2 bytes que identifican el comando, 4 bytes los datos y 2 bytes como terminador. Nos quedaría una trama de la siguiente forma:

Comando	Datos de Comando	Terminador
0x01, 0x56	0x24, 0x11, 0x87, 0x67	0x00, 0x00

Para enviarlo podríamos utilizar el siguiente código:

```
// Crear array con datos de Comando, Datos de Comando y Terminador.
new Comando[2], DatosComando[4], Terminador[2]

// Llenamos los arrays.
Comando [0] = 0x01
Comando [1] = 0x56
DatosComando [0] = 0x24
DatosComando [1] = 0x11
DatosComando [2] = 0x87
DatosComando [3] = 0x67
Terminador [0] = 0x00
Terminador [1] = 0x00
```



```
// Enviar array con los datos combinados.
I2C0MasterSendProtocolo(0xD0,Comando,2,DatosComando,4,Terminador,2,1)
```

Si queremos enviar solamente Comando[] y DatosComando[], hacemos "0" la cantidad de bytes a enviar para Terminador[].

```
// Enviar array los datos combinados.
I2C0MasterSendProtocolo(0xD0,Comando,2,DatosComando,4,Terminador,0,1)
```

Esta función puede ser muy útil para escribir protocolos o escribir datos en dispositivos con esquema de comandos.

7.3.2 Recepción de Datos

Las siguientes funciones reciben o leen datos desde un dispositivo I2C en el PLC.

I2CxMasterReceiveByte(Addr, &Byte=0): Recibe un byte del bus I2C número "x".		
Argumentos	Tipo	Descripción
Addr	E	Dirección I2C del dispositivo esclavo.
Byte	S	Variable donde se guarda el byte de dato a recibido / leído.
Retorno	Tipo	Descripción
0	S	Operación exitosa.
-1 a -20	S	Error, ver códigos retornados en Tabla 1 en página 6.
Notas		Descripción
1		La función genera las condiciones START, STOP y verificación de ACK requeridas por el bus automáticamente.

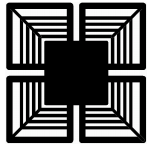
Ejemplo:

Leer un byte del dispositivo I2C en dirección 0xD0 utilizando interfaz I2C0:

```
// Variable para almacenar byte leído.
new Byte

// Realizar transacción I2C para leer byte.
I2C0MasterReceiveByte(0xD0, Byte)

// Comparar si el byte leído es menor a 50 y activar salida DOUT1.
if(Byte < 50)
{
    DoutSetOn(DOUT1)
}
```

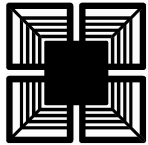


I2CxMasterReceive(Addr, Length=1, Data[]): Recibe un array de bytes del bus I2C número "x".		
Argumentos	Tipo	Descripción
Addr	E	Dirección I2C del dispositivo esclavo.
Length	E	Cantidad de bytes a leer desde el bus I2C (max. 128).
Data[]	S	Array donde se guardan los bytes de datos a recibidos / leídos.
Retorno	Tipo	Descripción
0	S	Operación exitosa.
-1 a -20	S	Error, ver códigos retornados en Tabla 1 en página 6.
Notas	Descripción	
1	La función genera las condiciones START, STOP y verificación de ACK requeridas por el bus automáticamente.	

Ejemplo:

Leer 5 bytes del dispositivo I2C en dirección 0xD0 utilizando interfaz I2C0:

```
// Array de 5 elementos para almacenar cada byte leído.  
new Registers[5]  
  
// Realizar transacción I2C para leer 5 bytes.  
I2C0MasterReceive(0xD0, 5, Registers)  
  
// Comparar si el primer byte leído es menor a 50 y activar salida DOUT1.  
if(Registers[0] < 50)  
{  
    DoutSetOn(DOUT1)  
}  
  
// Comparar si el segundo byte leído es mayor a 70 y activar salida DOUT2.  
if(Registers[1] > 70)  
{  
    DoutSetOn(DOUT2)  
}
```



7.3.3 Configuración de Bus I2C

Las siguientes funciones permiten configurar el bus I2C del PLC. **No deberían utilizarse** a menos que se sepa detalladamente lo que se está realizando. Se suministran para casos especiales.

I2CxInit(SpeedKHz): Inicializa completamente el hardware del bus I2C número "x".		
Argumentos	Tipo	Descripción
SpeedKHz	E	Velocidad en KHz a utilizar para el bus.
Retorno	Tipo	Descripción
0	S	Operación exitosa.
Notas		
1		El PLC antes de llamar a cualquier programa del usuario, inicializa completamente el hardware I2C. Por lo tanto no es necesario llamarla. Solo utilizar en casos especiales.
2		Si el bus I2C esta compartido con otros dispositivos utilizados por el PLC internamente, el setear con valores incorrectos la velocidad del bus puede provocar errores internos. Consulte en el foro de Slicetex por el valor correcto de velocidad para su modelo de PLC.

Ejemplo:

Inicializar la interfaz I2C0 con velocidad de 150 KHz:

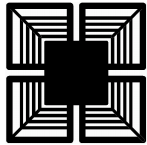
```
I2C0Init(150)
```

I2C0SetSpeed(SpeedKHz): Cambia la velocidad del bus I2C número "x".		
Argumentos	Tipo	Descripción
SpeedKHz	E	Nueva velocidad en KHz a utilizar para el bus.
Retorno	Tipo	Descripción
0	S	Operación exitosa.
Notas		
1		El PLC antes de llamar a cualquier programa del usuario, especifica una velocidad I2C. Por lo tanto no es necesario llamarla. Solo utilizar en casos especiales.
2		Si el bus I2C esta compartido con otros dispositivos utilizados por el PLC internamente, el setear con valores incorrectos la velocidad del bus puede provocar errores internos. Consulte en el foro de Slicetex por el valor correcto de velocidad para su modelo de PLC.

Ejemplo:

Configurar la interfaz I2C0 con velocidad de 150 KHz:

```
I2C0SetSpeed(150)
```



I2CxForceStandardSpeed(Force): Fuerza a utilizar una velocidad standard de bus I2C numero "x". Muy útil para emplear con dispositivos I2C que requieren velocidades menores a 100 KHz y combinarlos con dispositivos I2C que utilizan velocidades mayores a 100 KHz en el mismo bus.

Argumentos	Tipo	Descripción
Force	E	<p><u>Force = 1</u> Si la velocidad actual del bus es mayor a 100 KHz, esta función fuerza la velocidad del bus a 75 KHz (valor admitido en velocidad de modo Standard) y se guarda la velocidad actual en memoria. Si la velocidad actual del bus es menor a 100 KHz, no se ejecuta ninguna acción.</p> <p><u>Force = 0</u> Si la velocidad anterior del bus era mayor a 100 KHz, esta función recupera la última velocidad guardada y la establece en el bus. Si la velocidad anterior de era menor a 100 KHz, no se ejecuta ninguna acción.</p>
Retorno	Tipo	Descripción
0	S	Operación exitosa.
Notas		Descripción
1		Las especificaciones I2C nombran dos velocidades: Standard Mode (< 100 KHz) y Fast Mode (hasta 400 KHz). Algunos dispositivos soportan ambas y otros solo la Standard Mode.
2		Algunos dispositivos que solo soportan el Standard Mode son: DS1307 (RTC), LM75 (Sensor de temperatura), etc.

Ejemplo:

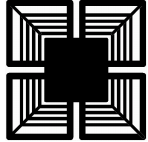
Supongamos querer utilizar un dispositivo I2C que solo soporta velocidad máxima de 100 KHz como el DS1307 (RTC), pero la interfaz I2C0 que utilizamos tiene conectados dispositivos funcionando a una velocidad de 150 KHz.

Entonces, lo ideal sería mantener la velocidad del bus I2C0 a 150KHz (más rápido) para comunicarnos con los dispositivos veloces y solo bajar la velocidad (a menos de 100 KHz) al intentar comunicarnos con el dispositivo DS1307 para leer la hora/fecha. Pero luego de leer la hora/fecha, restauramos la velocidad original del bus (150 KHz). Entonces:

```
DS1307_GetTime()
{
    new stat, new buf[7];

    // Asegurar que la velocidad del bus sea menor a 100 KHz.
    I2C0ForceStandardSpeed(1);

    // Seleccionar memoria del DS1307 para leer hora.
    if((stat=I2C0MasterSendByte(0xD0, 0))
    {
        // Restaurar velocidad del bus si fue cambiada.
        I2C0ForceStandardSpeed(0);
        return stat;
    }
}
```



```
// Obtener registros de hora del DS1307.
if((stat=I2C0MasterReceive(0xD0, 7, buf))
{
    // Restaurar velocidad del bus si fue cambiada.
    I2C0ForceStandardSpeed(0);
    return stat;
}

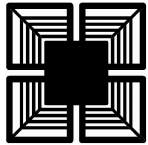
// Aplicar mascara a bits no utilizados.
buf[0] &= 0x7F; // Seconds.
buf[1] &= 0x7F; // Minutes.
buf[2] &= 0x3F; // Hours.
buf[3] &= 0x07; // Day of Week.
buf[4] &= 0x3F; // Date (Day of Month).
buf[5] &= 0x1F; // Month.
           // Year - not need.

// Con los valores de hora/fecha en buf[] podemos
// imprimirlos o guardarlos en una memoria antes de retornar.

// Restaurar velocidad del bus si fue cambiada.
I2C0ForceStandardSpeed(0);

// Return transaction status.
return stat;
}
```

Como vemos en el código anterior, la función Pawn **DS1307_GetTime()** se comunica con el DS1307 para leer fecha/hora, pero es independiente de la velocidad del bus utilizada. Si la velocidad del bus es menor a 100 KHz, nunca se cambia la velocidad. Pero si la velocidad del bus es mayor a 100 KHz, solo se cambia a una velocidad más baja al intentar comunicarse con el dispositivo DS1307, luego al salir de la función, se restaura la velocidad original del bus. La magia lo hace la función `I2CxForStandadSpeed()`.



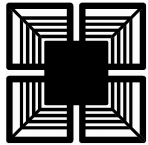
8 Abreviaciones y Términos Empleados

- **PLC:** Programable Logic Controller (Controlador Lógico Programable).
- **Modo PLC:** Permite programar el dispositivo mediante un programa en Ladder o Pawn.
- **I2C:** Bus para dispositivos llamado Inter-Integrated Circuit.

9 Historial de Revisiones

Tabla 2: Historia de Revisiones del Documento

Revisión	Cambios	Descripción	Estado
01 03/SEP/2015	1	1. Versión preliminar liberada.	Preliminar



10 Referencias

Ninguna.

11 Información Legal

11.1 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.

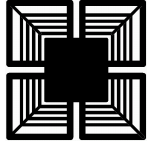
El derecho a realizar cambios: Slicetex Electronics se reserva el derecho de hacer cambios en la información publicada en este documento, incluyendo, especificaciones y descripciones de los productos, en cualquier momento y sin previo aviso. Este documento anula y sustituye toda la información proporcionada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



12 Información de Contacto

Para mayor información, visítenos en www.slicetex.com

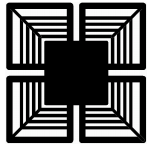
Para información, consultas y ventas, envíe un mail a: info@slicetex.com

Para soporte técnico ingrese a nuestro foro en: www.slicetex.com/foro

Ing. Boris Estudiez

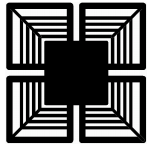
Slicetex Electronics
Córdoba, Argentina

© Slicetex Electronics, todos los derechos reservados.



13 Contenido

1	DESCRIPCIÓN GENERAL.....	1
2	LECTURAS RECOMENDADAS.....	2
2.1	EJEMPLOS.....	2
3	REQUERIMIENTOS.....	2
4	TEORÍA DE FUNCIONAMIENTO.....	3
5	CONEXIÓN AL PLC.....	4
6	COMPONENTES PARA LENGUAJE LADDER.....	5
7	FUNCIONES LENGUAJE PAWN.....	6
7.1	CONSIDERACIONES.....	6
7.2	TABLA DE CÓDIGOS I2C RETORNADOS.....	6
7.3	FUNCIONES I2C.....	7
7.3.1	TRANSMISIÓN DE DATOS.....	7
7.3.2	RECEPCIÓN DE DATOS.....	10
7.3.3	CONFIGURACIÓN DE BUS I2C.....	12
8	ABREVIACIONES Y TÉRMINOS EMPLEADOS.....	15
9	HISTORIAL DE REVISIONES.....	15
10	REFERENCIAS.....	16
11	INFORMACIÓN LEGAL.....	16
11.1	AVISO DE EXENCIÓN DE RESPONSABILIDAD.....	16
12	INFORMACIÓN DE CONTACTO.....	17
13	CONTENIDO.....	18
13.1	ÍNDICE DE TABLAS.....	19



13.2	ÍNDICE DE FIGURAS	19
-------------	--------------------------------	-----------

13.1 Índice de Tablas

Tabla 1: Códigos I2C retornados.....	6
Tabla 2: Historia de Revisiones del Documento.....	15

13.2 Índice de Figuras

Fig. 1: Conexionado típico I2C entre PLC y tres nodos esclavos.	3
Fig. 2: Conexión I2C	4

Copyright Slicetex Electronics 2015

www.slicetex.com