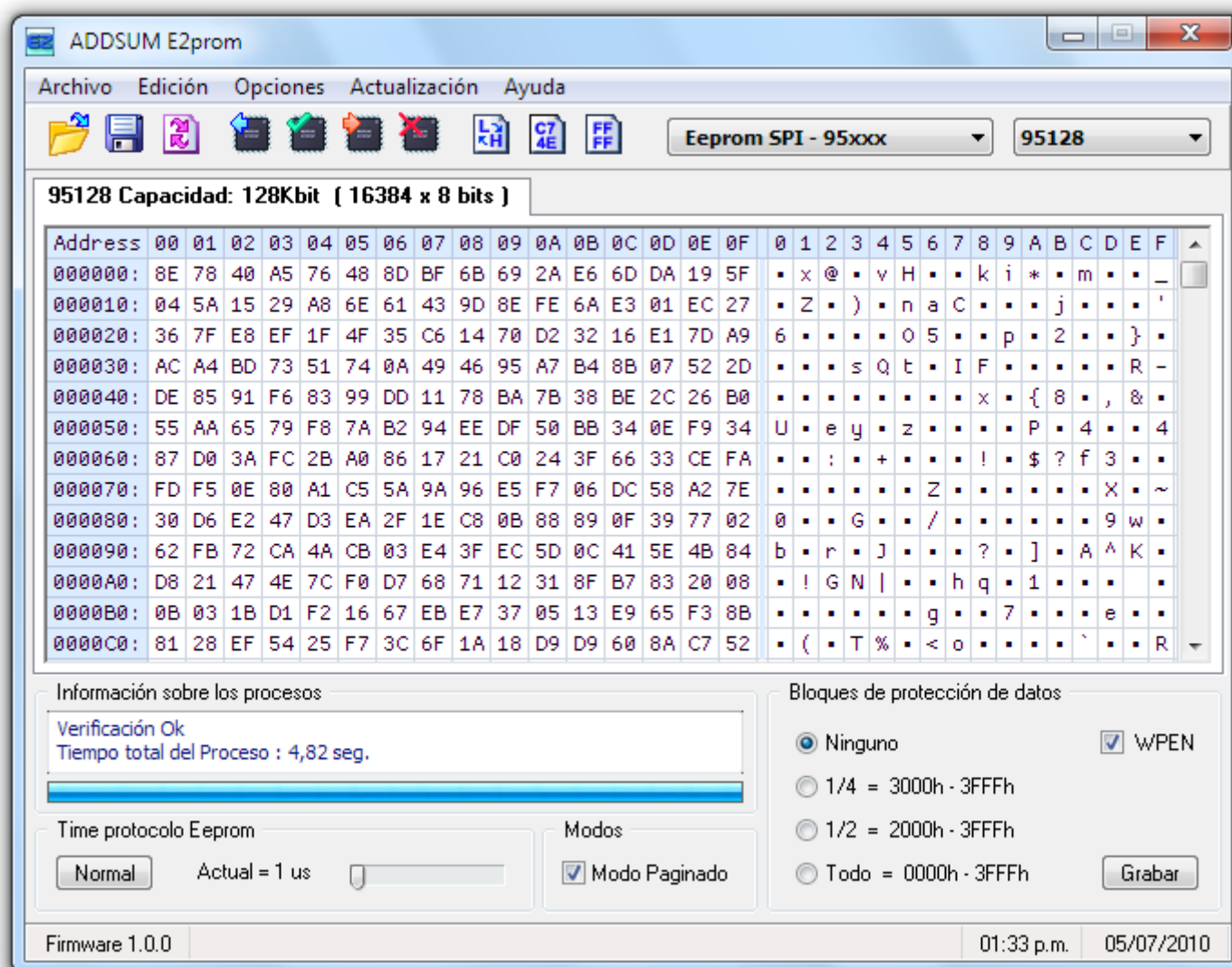


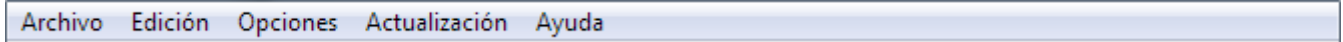
La interfaz de programación tiene la apariencia que se muestra en la Figura 1. De muy fácil manejo permite la programación de una amplia variedad de memorias Eeprom con diferentes protocolos de comunicación, I2C, Microwire, SPI, UNI/O. Sus controles y funciones están especificados en las siguientes secciones de este manual, se detallan además las conexiones con el programador, esquemas correspondientes, instalación y su operatividad en general.

Figura 1 – Interfaz de programación



La barra de menú -Figura 2-, despliega los diferentes ítems que a continuación se detallan:

Figura 2 – Barra de menú



Archivo:

- ▶ **Abrir Archivo:** importa un archivo para la programación. Son soportados los formatos de archivos hex y bin.
- ▶ **Guardar Archivo:** exporta un archivo de lectura con los datos del Buffer de memoria. El archivo es creado en el formato seleccionado (hex o bin). Si en la ruta especificada ya existe un archivo con el mismo nombre, será reemplazado por el nuevo archivo. Si éste es de sólo lectura, deberá guardarse con otro nombre.
- ▶ **Actualizar Archivo:** actualiza el archivo cargado en el Buffer de memoria.
- ▶ **Comprobar Checksum del Archivo Hex:** realiza una comprobación línea por línea del archivo Hex al momento de ser cargado en el Buffer de memoria para verificar que los datos sean correctos. Esta función no abarca los archivos con extensión .bin.
- ▶ **Archivos Recientes:** se muestran hasta los últimos diez archivos que han sido abiertos y cargados con sus respectivas rutas de directorio. Los archivos recientes pueden ser seleccionados para importarlos rápidamente. Observe que el historial de archivo comenzará estando en blanco en una instalación nueva hasta que sea importado un archivo. Cada archivo importado o seleccionado será cabecera de la lista.
- ▶ **Borrar Archivos Recientes:** borra todo el listado de archivos importados.
- ▶ **Salir:** cierra la aplicación.

Edición:

- ▶ **Limpiar Buffer de memoria:** el Buffer de memoria se reescribe con datos en blanco "FF" de acuerdo a la capacidad de memoria del dispositivo seleccionado.
- ▶ **Rellenar Buffer de memoria:** permite cargar espacios de memoria desde una dirección de inicio hasta una dirección final de acuerdo a la capacidad de memoria del dispositivo con un valor en formato Hexadecimal.
- ▶ **Intercambio de Bytes:** se intercambian los bytes en posiciones de memoria correlativas, es decir: el byte de la dirección 1 de memoria pasa a la dirección 2 y el byte de la dirección 2 pasa a la dirección 1 y así sucesivamente hasta completar la capacidad de memoria del dispositivo.
- ▶ **Buffer con datos aleatorios:** permite la carga del buffer de memoria con datos aleatorios, su uso se limita a la comprobación total de las celdas de memoria del dispositivo.
- ▶ **Habilitar cambios manuales en Buffer Grid:** habilita la introducción de datos manualmente en reemplazo de los existentes en la grilla o tabla de datos perteneciente al Buffer de memoria, ver: "*Buffer memoria de datos*".

Opciones:

- ▶ **Tipo de Fuente:** permite seleccionar entre tres tipos de fuentes mono espacio para visualizar el Buffer de datos.
- ▶ **Fuente Bold Buffer Grid:** selecciona la fuente en negrita.
- ▶ **Color Buffer Grid:** cambia la fuente a color azul.
- ▶ **Líneas Buffer Grid:** enmarca la tabla del buffer de memoria.

Actualización:

- ▶ **Grabar Firmware:** actualiza el Firmware (programa) del microcontrolador del programador Eeprom, directamente desde el Software. Con cada nueva actualización o revisión del programa podrá realizarse la actualización de forma simple y rápida desde la misma interfaz de usuario.

BARRA DE HERRAMIENTAS

La Figura 3 muestra la barra de herramientas con los diferentes controles de la interfaz de programación.

Figura 3 – Barra de herramientas



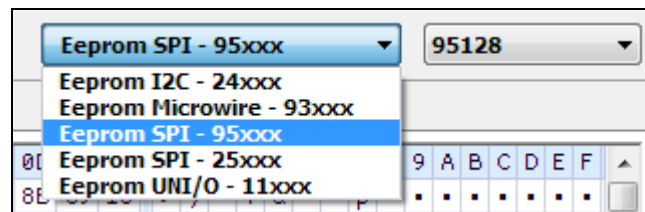
- ▶ **Abrir Archivo:** importa un archivo para la programación. Son soportados los formatos de archivos hex y bin.
- ▶ **Guardar Archivo:** exporta un archivo de lectura con los datos del Buffer de memoria. El archivo es creado en el formato seleccionado (hex o bin). Si en la ruta especificada ya existe un archivo con el mismo nombre, será reemplazado por el nuevo archivo. Si es requerido, utilice un nombre de archivo diferente para conservar el existente. Si el archivo es de sólo lectura, deberá guardarse con otro nombre.
- ▶ **Actualizar Archivo:** actualiza el archivo cargado en el Buffer de memoria.
- ▶ **Lectura de memoria:** lee la memoria Eeprom y carga el buffer con los datos leídos.
- ▶ **Verificación de memoria:** verifica la lectura de la memoria y compara con los datos almacenados en el Buffer de memoria de la aplicación.
- ▶ **Programación de memoria:** escribe el dispositivo con los datos cargados en el Buffer de memoria de la aplicación, la fuente de datos puede provenir de un archivo (hex o bin) o de una lectura de memoria.
- ▶ **Borrado de memoria:** borra los datos contenidos en la memoria Eeprom dejando el dispositivo en blanco (FF).
- ▶ **Intercambio de Bytes:** se intercambian los bytes en posiciones de memoria correlativas, es decir: el byte de la dirección 1 de memoria pasa a la dirección 2 y el byte de la dirección 2 pasa a la dirección 1 y así sucesivamente según la capacidad de memoria del dispositivo seleccionado.

- ▶ **Buffer con datos aleatorios:** permite la carga del buffer de memoria con datos aleatorios para programarlos en el dispositivo seleccionado; su uso se limita únicamente a la comprobación total de las celdas de memoria de la Eeprom.
- ▶ **Limpiar Buffer de memoria:** el Buffer de memoria se reescribe con datos en blanco "FF" de acuerdo a la capacidad de memoria del dispositivo seleccionado.

Cuadros de lista desplegable:

Seleccionan el tipo de memoria de acuerdo a las características y protocolo de comunicación y enlistan los dispositivos soportados por el programador -Figura 4-. Protocolos soportados: I2C, Microwire, SPI, UNI/O.

Figura 4 – Lista desplegable de tipos de memoria



- 1 - Seleccione en el cuadro desplegable el tipo de dispositivo por características y protocolo. Ej: SPI – 95xxx
- 2 - Seleccione en el siguiente cuadro desplegable la memoria por nomenclatura. Ej: 95128

BUFFER MEMORIA DE DATOS

El buffer de memoria de la aplicación almacena los datos cargados desde un archivo, desde una lectura previa de memoria o desde algún comando de edición. Los datos cargados se visualizan en el editor en forma de tabla con filas y columnas (Offset) tanto en formato hexadecimal (tabla izquierda) como en formato ascii (tabla derecha), tal como puede observarse en la Figura 5.

Los comandos de edición referentes al Buffer de memoria fueron detallados más arriba en las explicaciones correspondientes a "Barra de menú" y "Barra de Herramientas".

Figura 5 – Visualización de los datos del Buffer de memoria

25AA512 Capacidad: 512Kbit (65536 x 8 bits)																																						
Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F						
000000:	0B	F5	BD	22	F2	C5	0A	3D	E7	E5	A7	64	E9	58	95	DC	.	.	.	"	.	.	.	=	.	.	.	d	.	X	.	.						
000010:	81	D6	91	A5	25	EA	DD	BF	1A	0C	7B	E7	60	7D	6A	A3	%	{	.	`	}	j	.						
000020:	B3	FB	65	6D	9B	CB	B2	43	90	EC	50	AE	92	5E	FA	27	.	.	e	m	C	.	.	P	.	.	A	.	'					
000030:	2A	21	3A	EF	CD	F0	86	C6	C2	12	24	32	09	83	CE	AA	*	!	:	\$	2				
000040:	5C	03	0E	73	FF	16	5B	8D	F4	37	F8	B4	3B	A9	A2	2D	\	.	.	s	.	.	.	[.	.	7	-				
000050:	D1	28	E2	F6	76	F7	2F	11	6B	5C	CC	38	B1	8A	77	B0	.	(.	.	v	.	.	/	.	k	\	.	8	.	.	.	w	.				
000060:	04	4D	B6	7A	A8	1D	03	94	9D	3E	A0	BB	E3	AF	4B	77	.	M	K	w			
000070:	7A	72	8B	FC	1F	42	D7	18	14	63	75	82	5A	D4	20	FA	.	z	.	r	B	c	u	.	Z	.			
000080:	AC	53	5F	C4	51	67	AB	9A	46	88	05	06	8C	B5	F3	7E	.	S	_	.	Q	g	F	~			
000090:	DE	79	EF	48	C6	49	80	62	BC	69	D9	89	BE	DB	C8	02	.	y	.	H	.	I	.	.	b	.	i			
0000A0:	55	9E	C3	CA	F9	6E	54	E4	EE	8E	AD	0D	34	01	9C	84	U	n	T	4	.	.		
0000B0:	87	7F	98	4E	6F	93	E4	68	64	B3	82	8F	66	E1	70	08	.	.	.	N	o	f	.	p	
0000C0:	FD	A4	6C	D1	A1	74	B8	EB	96	95	56	57	DC	07	45	CF	.	.	.	l	V	W	.	E

Cambios manuales de datos:

La opción para poder realizar cambios manuales en el editor del Buffer de memoria debe estar habilitada desde el menú: *“Habilitar cambios manuales en Buffer Grid”*

- ▶ Seleccione con el puntero del mouse en el editor la celda que contiene los datos que desea modificar en la dirección de memoria representada por línea y columna (Offset). Puede ser tanto en el sector del editor de datos Hexadecimales o Ascii (caracteres del teclado).
- ▶ Introducir el nuevo dato. Para el sector Hex sólo es posible el formato Hexadecimal, serán admitidos los números 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 y letras A, B, C, D, E, F (00-FF hexadecimal). Para el sector Ascii será admitido este formato (ascii) correspondiente a los caracteres de teclado 0-127 (decimal).
- ▶ Una vez reemplazado el dato, quedará almacenado en el Buffer de memoria en la misma dirección que el dato anterior y automáticamente se incrementará la selección hacia la celda siguiente. El usuario puede o no continuar introduciendo datos.
- ▶ Para borrar caracteres de la celda utilizar la tecla de retroceso. Una celda con datos incompletos será reemplazada con el dato original correspondiente a ese direccionamiento de memoria.

Nota: Si existe un error durante la verificación de datos grabados en la memoria Eeprom y los cambios manuales del Buffer de memoria están habilitados desde el menú, quedará seleccionada en el editor la celda con la dirección del dato que produjo el error durante la escritura.

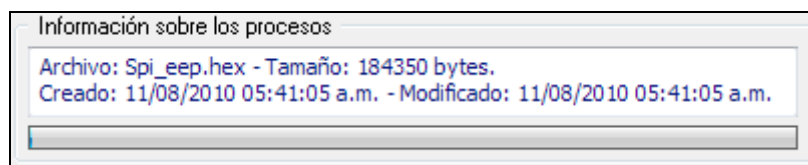
INFORMACIÓN SOBRE LOS PROCESOS

Ventana de estado:

Esta ventana muestra información y el estado de las operaciones en progreso.

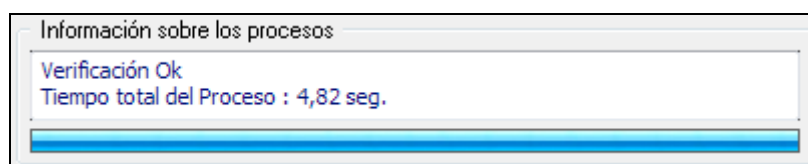
- Durante la carga de un archivo al Buffer de memoria, informa: nombre de archivo, tamaño en bytes, fecha-hora de creación y fecha-hora de última modificación –Figura 6–.

Figura 6 – Información mostrada durante la carga de un archivo



- Durante procesos como lectura, verificación, escritura, borrado, informa: operaciones en progreso, resultado y tiempo involucrado en el proceso –Figura 7–.

Figura 7 – Información mostrada durante los procesos



- Si existe un error durante la verificación de datos grabados en la memoria Eeprom, el texto será expuesto en color rojo informando: la dirección por fila y columna donde se produjo el error, el dato verificado desde la aplicación y el dato leído desde la memoria Eeprom.
- Durante la actualización del Firmware del programador, informa sobre la operación en curso y si la actualización fue completada con éxito.

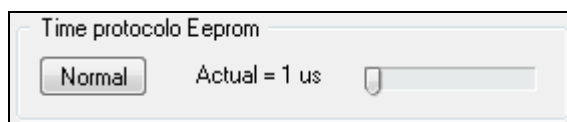
Barra de progreso:

Muestra el progreso de una operación.

TIME DEL PROTOCOLO EEPROM

Los tiempos de los protocolos soportados por el programador de memorias Eeprom pueden ser controlados y modificados linealmente mediante el control deslizante diseñado para ello en la aplicación, como muestra la Figura 8. Dicho control permite variar el tiempo establecido aumentando el “time” del protocolo de comunicación para hacerlo más lento y de esta forma poder ejecutar las operaciones en forma exitosa si un dispositivo no responde adecuadamente dentro de los márgenes normales de funcionamiento.

Figura 8 – Control deslizante para el tiempo de protocolo



En algunos dispositivos, en el caso de memorias con protocolo de comunicación I2C, puede variarse el “time” para que sea más rápido, siempre y cuando la memoria soporte la velocidad de comunicación establecida.

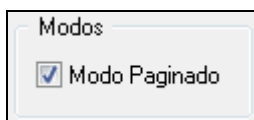
El botón “Normal” reestablece los valores normales de origen por defecto.

MODOS

Modo paginado:

El modo paginado –Figura 9- abarca a los dispositivos que soportan la escritura por página de memoria, es decir, que pueden ser programados varios bytes de datos en un mismo tiempo, acelerándose el proceso de escritura. El modo paginado incluye a los dispositivos con protocolo I2C, SPI, UNI/O.

Figura 9 – Selección del Modo Paginado



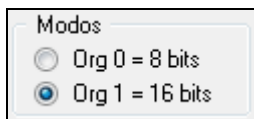
El modo paginado está seleccionado por defecto en el Software del programador mediante el casillero con tilde activado. Para una programación de datos byte por byte (modo no paginado) destildar dicho casillero.

Nota: este modo es para transferencias de datos y no involucra ninguna señal de control hacia alguno de los pines del dispositivo.

Modo organización 8 – 16 Bits:

El modo de organización de 8 y 16 bits, abarca a los dispositivos con protocolo Microwire -Figura 10-. Según el fabricante y nomenclatura del dispositivo, puede soportar la organización de programación en 8 y/o 16 bits.

Figura 10 – Selección del Modo de organización 8-16 bits



Es importante en estos dispositivos conocer, por medio de los Data Sheet del fabricante, qué formato de organización soportan. Si el dispositivo está programado con una organización de 8 bits, debe leerse en 8 bits. Lo mismo para una organización de 16 bits.

Por defecto el Software de programación selecciona a los dispositivos con letra “A” en 8 bits y “B” en 16 bits, esto **NO** significa un estándar generalizado entre fabricantes de memorias Eeproms, hay dispositivos “A” que también soportan el formato de 16 bits y “B” que soportan el formato de 8 bits.

El usuario debe informarse de los datos que brinda el fabricante del dispositivo en todos los casos.

Nota: El modo organización 8 - 16 bits incluye al pin “Org” en los dispositivos Microwire 93xxx. El programador suministra hacia ese pin una tensión positiva (nivel alto) en caso de ser seleccionada una organización de escritura por 16 bits o una tensión próxima a GND (nivel bajo) si fue seleccionada una organización por 8 bits de escritura desde la aplicación; esto es adecuado para la programación individual de la memoria desde un zócalo, facilitando un mínimo de conexionado a tener en cuenta.

CONFIGURACION DE DISPOSITIVOS

Desde el software de la aplicación es posible la configuración en algunos tipos de dispositivos Eeprom. Ver “Hardware”.

BARRA DE ESTADO

En la Figura 11 se muestran los contenidos de la Barra de Estado. El primer panel muestra la versión del Firmware del programador (programa grabado en el hardware del equipo). Si no existe Firmware debe visualizarse la versión del Bootloader (cargador del Firmware).

El segundo panel muestra la ruta del archivo cargado al Buffer de memoria de la aplicación. Si dicha ruta excede los 64 caracteres será mostrada en formato corto.

Los paneles tercero y cuarto, muestran hora y fecha del sistema respectivamente.

Figura 11 – Barra de Estado



HARDWARE

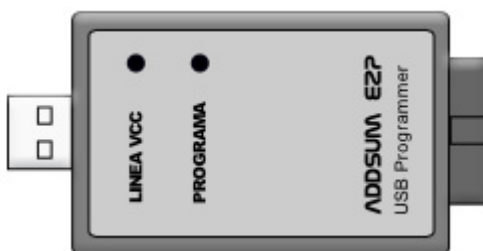
El hardware del programador basado en un microcontrolador con módulo USB integrado permite un mínimo de componentes externos, reducción de tamaño del circuito y establece una comunicación rápida con el Software de aplicación. La salida de señales por el conector es totalmente reconfigurable automáticamente de acuerdo al dispositivo seleccionado y su protocolo de comunicación, permitiendo realizar las operaciones de programación, lectura, verificación, etc., desde un mismo y único zócalo para todos los dispositivos soportados. El Firmware del programador es actualizable con nuevas versiones o revisiones. La Figura 12 muestra las características del programador.

Figura 12 – Características del programador



Conector del programador –Figura 13-:

Figura 13

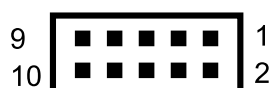


Salida conector IDC para cable plano

- Pin superior = 1, inferior = 2
- Pin superior = 3, inferior = 4
- Pin superior = 5, inferior = 6
- Pin superior = 7, inferior = 8
- Pin superior = 9, inferior = 10

La Figura 14 es una vista frontal del conector IDC del programador de diez contactos con la numeración correspondiente de pines.

Figura 14



SALIDAS

Pines **1, 2, 3, 4, 5, 6, 7, 8** corresponden a señales de control, datos y reloj reconfigurables automáticamente según memoria Eeprom y protocolo de comunicación correspondiente.

Pins **9, 10** corresponden a una salida de tensión de 5 volts positivos suministrados por el puerto Usb de la PC a través del conector del programador, está disponible en caso de ser requerida una tensión suplementaria para alimentar una placa de circuito. El límite de corriente es de 100 mA. Para la programación de memorias esta tensión secundaria no se utiliza.

Pin **9**, +5 volts (Vbus).

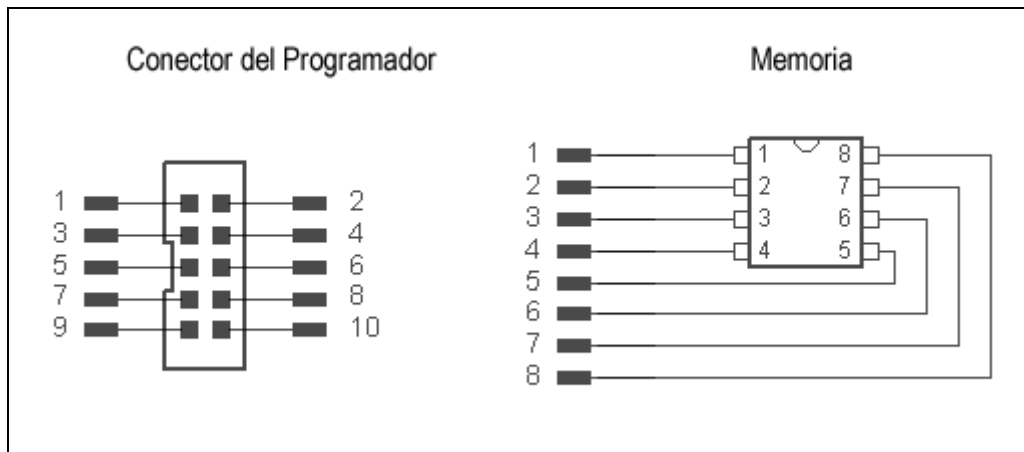
Pin **10**, negativo (Gnd).

CONEXIONADO DE MEMORIAS

El conexionado universal entre programador y memoria se realiza como muestra la Figura 15. La interfaz de comunicación entre el programador y los diferentes tipos de dispositivos soportados es común en todos los casos y comparten la misma distribución de pines; el programador automáticamente reconfigura el puerto de salida de acuerdo al tipo de memoria y protocolo de comunicación utilizado, es decir: señales de control, datos, reloj, Vcc y Gnd necesarias se reconfiguran automáticamente para utilizar el mismo conexionado general de acuerdo al tipo de dispositivo seleccionado.

Observe que la interfaz de conexión entre el conector del programador y los pines del dispositivo Eeprom comparten la misma numeración: 1/1, 2/2, 3/3, etc.

Figura 15 – Conexionado de las memorias



Se describen a continuación los detalles de conexionado entre la interfaz del programador y los dispositivos.

Importante: cuando sea necesario programar, leer, verificar, etc., dispositivos Eeprom que se encuentren ya conectados en una placa de circuito, deben ser primero desmontados o retirados de dicha placa en todos los casos y sin excepción. Esto significa que las tareas a realizar con el programador y memoria deberán ser efectuadas en forma independiente y no en circuito. Caso contrario se correría el riesgo de:

- Atenuación de las señales de datos y reloj.
- Consumo excesivo en el puerto del conector del programador por baja impedancia hacia masa o positivo de algún pin.
- Conflicto con otros componentes del circuito (microcontroladores que manejan dicha Eeprom, etc.).
- Lecturas o programaciones erróneas.

- Deterioro del programador/puerto Usb por señales en corto o sobreconsumo por mala conexión.

En general las placas de circuito con memorias Eeprom no están preparadas para soportar una programación o reprogramación en circuito, por este motivo no se aconseja emplear este método y en estas circunstancias no se garantiza el correcto funcionamiento del equipo.

Protocolo I2C memorias 24xxx:

La Figura 16 muestra la posible configuración de direccionamiento en el hardware para la serie de memorias 24xxx, a través de las señales de control sobre los pines A0, A1, A2 y su posible mapeo desde el Software del programador.

En la mayoría de las ocasiones escribir la memoria con determinada configuración no será necesario, solamente en los casos en los que más de un dispositivo comparta el mismo bus I2C y sea requerido un direccionamiento especial.

En el Software del programador, por defecto la habilitación está desactivada. Si no requiere un direccionamiento en el bus I2C de la aplicación, no habilite la selección de pines.

Figura 16 – Protocolo I2C para memorias 24xxx

Dirección dispositivo en el Hardware

A2

A1 Actual = 000

A0

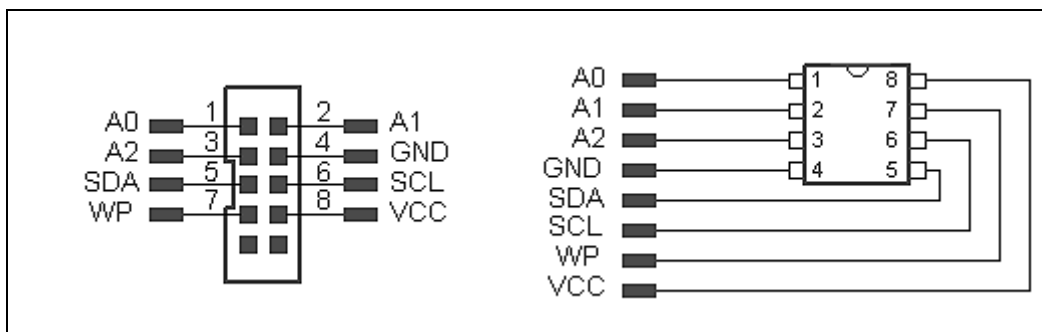
Habilitar selección de pines

Nota: la configuración de direccionamiento por hardware desde la aplicación suministra la salida de señales de control desde el conector del programador hacia la memoria sobre los pines A0, A1, A2 (1, 2, 3).

Tabla 1 – Eeprom I2C:

Nº de pin	Señal	Descripción
1 – 2 – 3	A0 – A1 – A2	Entrada de direccionamiento del dispositivo en el bus I2C
4	GND	Negativo (masa)
5	SDA	Línea I/O serial bidireccional de datos
6	SCL	Línea serial de reloj
7	WP	Protección contra escritura. Habilita con nivel bajo (Gnd)
8	VCC	Tensión de alimentación de la memoria

Figura 17: Diagrama de pines y sus respectivas señales memorias 24xxx



Nota: la resistencia requerida hacia pull-up, principalmente en la línea de datos propia del protocolo I2C, es proporcionada internamente por el programador facilitando de esta forma el mínimo conexionado.

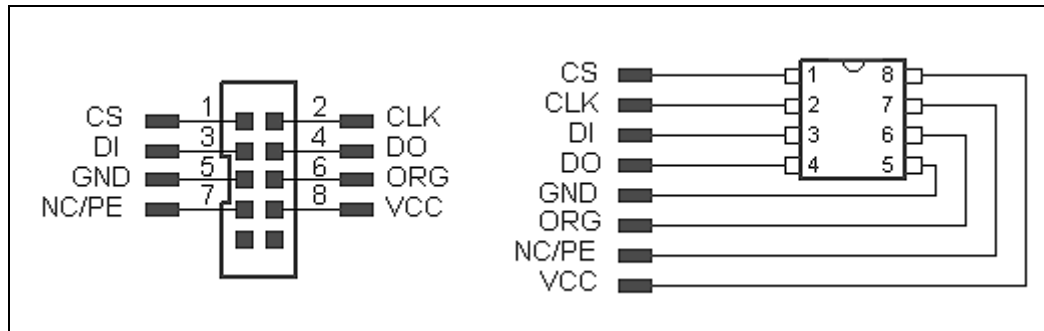
Protocolo Microwire memorias 93xxx:

Ver ítem “Modos” > “Modo organización 8 – 16 Bits” para su configuración por software.

Tabla 2 – Eeprom Microwire:

Nº de pin	Señal	Descripción
1	CS	Selección de chip
2	CLK	Línea serial de reloj
3	DI	Línea serial de entrada de datos
4	DO	Línea serial de salida de datos
5	GND	Negativo (masa)
6	ORG	Organización 8 / 16 bits
7	NC/PE	No conexión o programación habilitada (enable) en 93x76/x86
8	VCC	Tensión de alimentación de la memoria

Figura 18: Diagrama de pines y sus respectivas señales memorias 93xxx



Protocolo SPI memorias 25xxx – 95xxx:

Registro de estado: un registro interno de la memoria que permite la configuración para la protección contra escritura. Se describen a continuación los detalles para su utilización:

Protección contra Escritura: permite al usuario seleccionar uno de los cuatro niveles de protección para la memoria, mediante la escritura de los bits apropiados en el registro de estado. La memoria está dividida en cuatro segmentos. La partición es controlada desde el Software de la siguiente manera: ninguna protección, $\frac{1}{4}$ capacidad de memoria protegida, $\frac{1}{2}$ capacidad de memoria protegida, protección total de la memoria –ver Figura 19-. El direccionamiento de inicio y final del block protegido contra escritura es visualizado en formato hexadecimal de acuerdo a la capacidad de memoria del dispositivo. Asimismo, una lectura del dispositivo indica cuáles bloques se hallan actualmente protegidos contra escritura.

WPEN: “Write Protect Enable”, si WPEN está sin tilde, configuración en “0” (estado por defecto en el Software), permite la escritura en el registro de estado de los bloques de protección, sin importar el estado o nivel (alto o bajo) del pin WP (Pin 3) de la memoria (protección de escritura), ver Tabla 3.

Si WPEN está con tilde, configuración en 1, la posibilidad de escribir en el registro de estado depende del nivel lógico que esté presente en el pin WP:

- Si el pin WP está a nivel alto, es posible escribir en el registro de estado los bloques de protección.
- Si el pin WP está a nivel bajo, cualquier intento de modificar el registro de estado será ignorado por el dispositivo.

Nota: por defecto las configuraciones y líneas de señales por hardware están dispuestas para permitir la escritura y modificación del registro de estado del dispositivo.

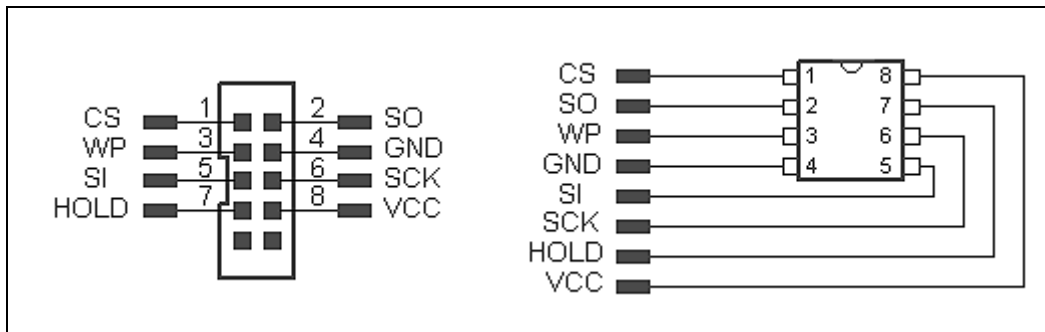
Figura 19 – Partición de la protección de datos

El botón “Grabar” permite la programación de la configuración en el registro de la memoria.

Tabla 3 – Eeprom SPI:

Nº de pin	Señal	Descripción
1	CS	Selección de chip
2	SO	Línea serial de salida de datos
3	WP	Protección contra escritura
4	GND	Negativo (masa)
5	SI	Línea serial de entrada de datos
6	SCK	Línea serial de reloj
7	HOLD	Pausa
8	VCC	Tensión de alimentación de la memoria

Figura 20: Diagrama de pines y sus respectivas señales memorias 25xxx/95xxx

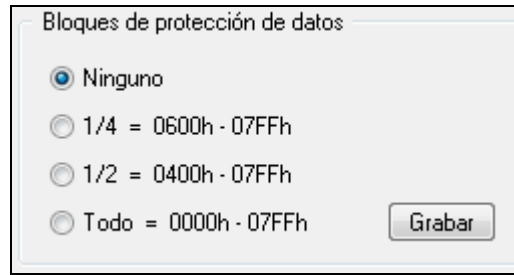


Protocolo UNI/O memorias 11xxx :

Registro de estado: registro interno de la memoria que permite la configuración para la protección contra escritura. Se describen a continuación los detalles para su utilización:

Protección contra Escritura: permite al usuario seleccionar uno de los cuatro niveles de protección para la memoria, mediante la escritura de los bits apropiados en el registro de estado. La memoria está dividida en cuatro segmentos. La partición es controlada desde el Software de la siguiente manera: ninguna protección, ¼ capacidad de memoria protegida, ½ capacidad de memoria protegida, protección total de la memoria –ver Figura 21-. El direccionamiento de inicio y final del block protegido contra escritura es visualizado en formato hexadecimal de acuerdo a la capacidad de memoria del dispositivo. Asimismo, una lectura del dispositivo indica cuáles bloques se hallan actualmente protegidos contra escritura.

Figura 21 – Partición de la protección de datos

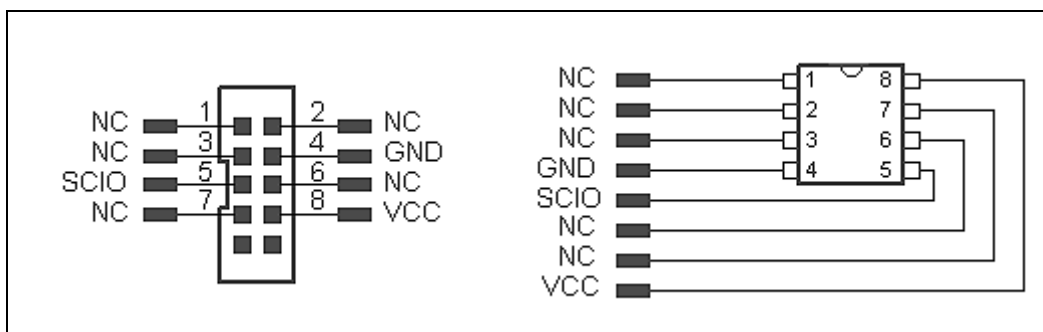


El botón “Grabar” permite la programación de la configuración en el registro de estado de la memoria.

Tabla 4 – Eeprom UNI/O:

Nº de pin	Señal	Descripción
1-2-3-6-7	NC	No conectados internamente en el chip
4	GND	Negativo (masa)
5	SCIO	Línea I/O serial bidireccional de datos
8	VCC	Tensión de alimentación de la memoria

Figura 22: Diagrama de pines y sus respectivas señales memorias 11xxx



Nota: el conexionado explicado y mostrado en todos los ejemplos corresponde a encapsulados de memoria PDIP, SOIC, TSSOP. Para otro tipo de packaging de dispositivos, deberá remitirse al conexionado del fabricante para realizar los cambios correspondientes en la interfaz de programación del programador.

INSTALACION

- ▶ Haga correr el Setup de la aplicación del programador, siguiendo los pasos del asistente hasta finalizar. La instalación es rápida y sencilla.
- ▶ Conecte el cable USB entre el programador y un puerto de la PC. Inmediatamente el Hardware debe ser reconocido por el sistema operativo dando inicio a la instalación automática de los drivers necesarios y a su correspondiente enumeración dentro del sistema. En las Figuras 23 y 24 puede observarse el proceso de instalación señalado, los mensajes pueden variar de acuerdo al sistema operativo en uso.

Figura 23 – Indicador de la instalación del software

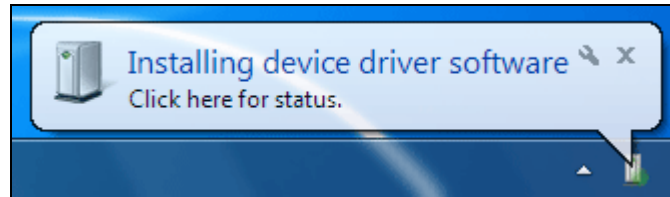
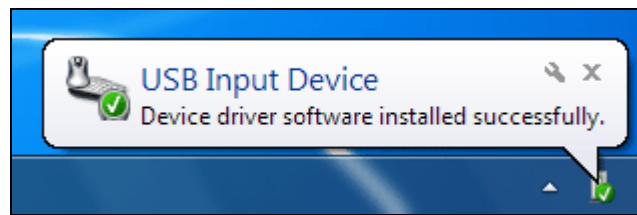


Figura 24 – Indicador de que la instalación ha sido exitosa



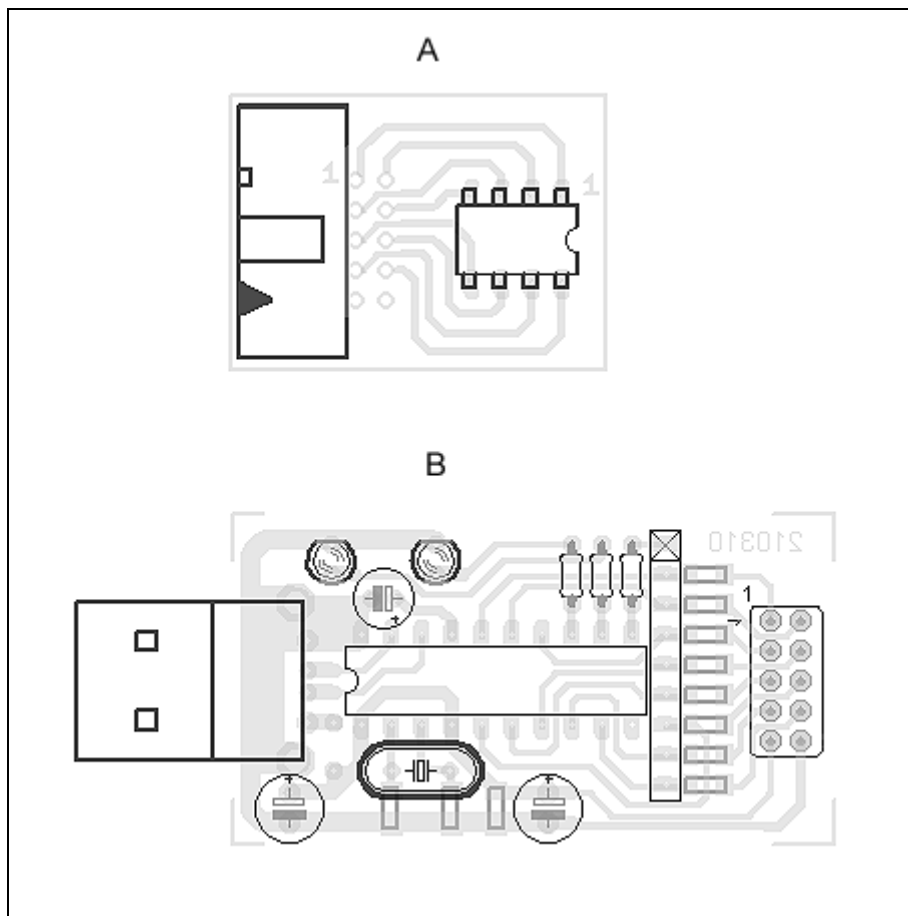
- ▶ Abra el Software de aplicación y verifique el correcto funcionamiento del equipo realizando alguna operación con una memoria Eeprom.

Nota: El cable de conexión USB debe ser de buena calidad, apto para trabajar con Usb 2.0. Un cable de dudosa calidad puede aparejar problemas de programación, comunicación, etc.

La figura 25-A muestra la típica placa (PCB) para programación de Eeproms desde un zócalo. El conector IDC de 10 contactos para cable plano, permite una conexión directa y pin a pin con el programador para los diferentes dispositivos y protocolos soportados por el equipo.

La figura 25-B, muestra la placa de circuito del hardware del programador de memorias Eeprom.

Figura 25 – PCB



En la figura 26 se puede observar la salida desde el conector IDC del programador. Los conductores 1 a 8 corresponden a los pines 1 a 8 de los diferentes tipos de Eeprom soportadas.

Los conductores 9 (+) y 10 (Gnd) corresponden a una salida de tensión auxiliar de 5 volts directamente de la línea Vbus del conector USB. La corriente máxima soportada es de 100 Ma. Esta tensión auxiliar no es manejada para la programación, puede ser utilizada si es necesaria una tensión auxiliar para alimentar un circuito de aplicación con un consumo menor a 100 Ma. Tenga precaución evitando cortos o un excesivo consumo en esta línea auxiliar.

Figura 26 - Conector

