

Confidencial

EPSON

LIBRERIA DE COMUNICACION BAJO NIVEL

ESPECIFICACIONES

IMPRESORAS FISCALES

ESTANDAR	
Rev. Nro.	1.4
Notas	

Fecha de copia	26/09/2006
Copiado por	




EPSON ARGENTINA S.A.
SEIKO EPSON CORPORATION

HOJA DE REVISIONES

Rev.	HOJA	CAMBIOS
1.0	Todas	Revisión Inicial. Versión de la librería 1.00
1.1	Todas	Corrección de parámetros en funciones. Versión de la librería 1.02.
1.2	Todas	Cambio en definición de función GetAPIVersion(). Versión de la librería 1.03.
1.3	Todas	Agregado de librería .Lib para Win32. Versión de la librería 1.03.
1.06	Todas	Cambio de esquema de versiones de documentación. Corrección en time-outs utilizando adaptadores del tipo RS232-USB Upgrade a versión de librería 1.06.
1.4	Todas 7	Nueva version de librería 1.4.0.0. Problemas con tramas con datos al máximo permitido corregido. Corrección de ejemplo de declaración en C++ para la función OpenSerialPort.
<div>TITULO</div> <div>LIBRERIA DE COMUNICACION BAJO NIVEL</div> <div>ESPECIFICACIONES</div> <div>IMPRESORAS FISCALES</div>		

Confidencial

Nomenclatura Utilizada en este Documento

Símbolo	Significado
	Este símbolo indica que el texto que seguirá hace referencia a otros temas de este documento.
	Este símbolo indica que el texto que seguirá indica cómo se puede utilizar la librería.
	Este símbolo indica que el texto que seguirá describe los cuidados que se deben tomar en cuenta en la utilización de las funciones de los ejemplos dados.

Nomenclatura de la librería para Win 32

Nombre	Significado
<Nombre_Librería>.dll	La librería para Win 32 usa el recurso de Dynamic Link Library.

Nomenclatura de la librería .LIB para Win 32

Nombre	Significado
<Nombre_Librería>.lib	La librería estática para Win 32.

Nomenclatura de la librería para Linux

Nombre	Significado
<Nombre_Librería>.a	El nombre de la librería con extensión "a", indica que la librería es del tipo estática.
<Nombre_Librería>.so	El nombre de la librería con extensión "so", indica que la librería es del tipo dinámica.

Nomenclatura de la librería para DOS

Nombre	Significado
<Nombre_Librería><S>.lib	El nombre de la librería con terminación "S" indica que la librería usa modelo de compilación Small .
<Nombre_Librería><L>.lib	El nombre de la librería con terminación "L" indica que la librería usa modelo de compilación Large .

TABLA DE CONTENIDOS

Capítulo 1 - Introducción	1
Capítulo 2 - Funciones de la Librería.....	2
2.1 Funciones de Configuración y Control del Puerto Serie	5
2.1.1 OpenSerialPort.....	6
2.1.2 CloseSerialPort	8
2.1.3 SendSerialMessage	10
2.1.4 ResendSerialMessage	12
2.2 Funciones de Manejo del Buffer de Salida	14
2.2.1 AddMessageField	15
2.2.2 AddMessageFieldEx	19
2.2.3 PurgeMessage	24
2.3 Funciones de Manejo del Buffer de Entrada	25
2.3.1 Para sistemas Win 32 e Linux	25
2.3.2 Para sistema Dos.....	26
2.3.3 GetMessageField	27
2.3.4 GetMessageFieldEx.....	31
2.3.5 GetFieldCounter	35
2.4 Funciones de Status	37
2.4.1 GetCommunicationState	38
2.4.2 GetLastCommError	40
2.5 Funciones Informativas.....	42
2.5.1 GetApiVersion	43
2.5.2 GetSentFrame.....	45
2.5.3 GetReceivedFrame	47
Capítulo 3 - Guías de como Compilar y Linkear en Linux.....	49
3.1 Compilar y Linkear	49
3.1.1 Comando del GCC – Librería estática	50
3.1.2 Comando del GCC – Librería dinámica	50

Capítulo 1 - Introducción

La librería de bajo nivel desarrollada por EPSON encapsula internamente todo el protocolo de comunicación de las Impresoras fiscales EPSON. Esto garantiza una mayor agilidad en el desarrollo de aplicativos y mayor seguridad en la comunicación con la impresora dado que la misma ha pasado por rigurosos procedimientos internos de testing.

Esta librería fue desarrollada para permitir mayor portabilidad de modo de poder ser compatible con los diversos sistemas operativos. En principio esta librería contempla la utilización de esta en los siguientes sistemas y modelos de utilización:

- ✓ Win 32 – Dynamic Link Library (DLL)
- ✓ Win 32 – Static Library (Lib)
- ✓ Dos – Small and Large Library (Lib)
- ✓ Linux – Static and Dynamic Library

Para ambientes Win 32 y Linux la librería posee un funcionamiento interno "Multithread" , es decir, que las funciones no bloquean el flujo de la aplicación que utiliza sus funciones pudiendo mientras tanto hacer otras tareas. La impresora y la librería realizan operaciones en background.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 1

Capítulo 2 - Funciones de la Librería

Las funciones de la librería fueron definidas siguiendo los siguientes prototipos:

DWORD **Nombre_de la Función (...)**
VOID **Nombre_de la Función (...)**
BYTE * **Nombre_de la Función (...)**

La siguiente tabla define los tipos de datos utilizados en los retornos de los parámetros de las funciones de la librería con su respectivo tamaño en bits.

Tipos	Descripción	Ejemplo de declaración en C/C++
byte	Variable de 8 bits	unsigned char
word	Variable de 16 bits	unsigned short
dword	Variable de 32 bits	unsigned int
bool	0 (FALSE) ó ≠ 0 (TRUE)	int/bool

La siguiente tabla describe los posibles códigos de error retornados por las funciones. Estos errores fueron divididos en categorías para facilitar la identificación de los mismos.

Constante	Valor (Hex)	Descripción del Retorno
EFPROT_SUCCESS	0 (0000)	Operación efectuada con éxito.
Errores del 1 a 5: Error de Apertura y/o Configuración del Puerto Serie		
EFPROT_PORT_ALREADY_OPEN	1 (0001)	Puerto Serial abierto.
EFPROT_PORT_IN_USE	2 (0002)	Puerto Serial en uso por otra aplicación.
EFPROT_PORT_NOT_OPEN	3 (0003)	Puerto Serial Cerrado
EFPORT_INVALID_COM	4 (0004)	Puerto Serial utilizado en la apertura inválido o inexistente.
EFPROT_INVALID_BAUD_RATE	5 (0005)	Velocidad del Puerto Serial inválido o no soportada.
Errores de 6 a 8: Errores de Envío de Datos		
EFPROT_TIMEOUT_ERROR	6 (0006)	Time-out de comunicación.
EFPROT_SEND_ERROR	7 (0007)	Error en envío de datos.
EFPROT_OFF_OR_DISCONNECTED	8 (0008)	Impresora desconectada u offline.
Error 9: Error de agregado de datos en el frame de envío		
EFPROT_BUFFER_OVERFLOW	9 (0009)	Límite del buffer de envío de datos alcanzado.

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 2

Confidencial

Errores del 10 a 13: Errores en la recuperación/agregado de datos en el buffer de envío/respuesta.

EFPROT_FIELD_INVALID	10 (000A)	Formato del campo de entrada al buffer de comandos inválido.
EFPROT_ASWR_INVALID	11 (000B)	Campo de respuesta solicitado inválido.
EFPROT_ASWR_FIELD_INEXISTENT	12 (000C)	Campo de respuesta inexistente.
EFPROT_FIELD_INVALID_TYPE	13 (000D)	Tipo de campo de respuesta inválido.

Error 14: Error interno

EFPROT_INVALID_PARAM	14 (000E)	Parámetro pasado función inválido.
----------------------	-----------	------------------------------------

Error 15: Error interno

EFPROT_INTERNAL_ERROR	15(000F)	Error interno de la Librería.
-----------------------	----------	-------------------------------

A continuación, se definen los tipos de variables que utilizan las funciones de agregado y captura de datos del buffer de la impresora.

Constante	Valor (Hex)	Descripción
-----------	-------------	-------------

Constantes para el Agregado de Datos

EFPROT_TYPE_BINARY	1 (0001)	Indica que los valores informados están todos en formato binario.
EFPROT_TYPE_ASCII	2 (0002)	Indica que los valores informados están todos en formato ASCII.

Constantes para a Recepción de Datos con Tipos Específicos

EFPROT_TYPE_UCHAR	0 (0000)	Campo numérico de 8 bits sin signo (0 a 255)
EFPROT_TYPE_CHAR	1 (0001)	Campo numérico de 8 bits con signo (-128 a +127)
EFPROT_TYPE_USHORT	2 (0002)	Campo numérico de 16 bits sin signo (0 a 65.535)
EFPROT_TYPE_SHORT	3 (0003)	Campo numérico de 16 bits con signo (-32.768 a 32.767)
EFPROT_TYPE_UINT	4 (0004)	Campo numérico de 32 bits sin signo (0 a 4.294.967.295)
EFPROT_TYPE_INT	5 (0005)	Campo numérico de 32 bits con signo (-2.147.483.648 a 2.148.483.647)
EFPROT_TYPE_ULONG	6 (0006)	Campo numérico de 32 bits sin signo (0 a 4.294.967.295)
EFPROT_TYPE_LONG	7 (0007)	Campo numérico de 32 bits con signo (-2.147.483.648 a 2.148.483.647)
EFPROT_TYPE_STRING	8 (0008)	Vector de caracteres.

Constantes para Posicionamiento de los Datos en el Buffer de Respuesta

EFPROT_PRT_ANSWER	1 (0001)	Retorna el Status de la Impresora después de la ejecución de un comando.
-------------------	----------	--

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 3

Confidencial

EFPROT_FSC_ANSWER	2 (0002)	Retorna el Status Fiscal después de la ejecución de un comando.
EFPROT_RSV_ANSWER_1	3 (0003)	Campo Reservado.
EFPROT_CMD_ANSWER	4 (0004)	Retorna la respuesta del comando enviado a la impresora.
EFPROT_RSV_ANSWER_2	5 (0005)	Campo Reservado.
EFPROT_ASW_FIELD_XX	XX (XXXX)	Parámetros de Salida del Comando.

Constantes para Solicitar Status de la Librería

EFPROT_CLOSED	0 (0000)	Comunicación Serial Cerrada.
EFPROT_IDLE	1 (0001)	Librería en estado idle.
EFPROT_BUSY	2 (0002)	Librería efectuando tareas en background.

Las constantes EFPROT_ASW_FIELD_XX indican los campos de respuesta que pueden recibirse en respuesta a un comando. Estos valores pueden variar desde 0 a n, siendo 6 el valor del primer campo.

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 4

2.1 Funciones de Configuración y Control del Puerto Serie

Las funciones de este bloque son utilizadas para la configuración, apertura y cierre del puerto serie como también el envío de datos a través de la misma. Estas son:

Función	Descripción
OpenSerialPort	Efectúa la apertura del puerto serie, allocating los recursos de software y hardware necesarios para su utilización.
CloseSerialPort	Efectúa el cierre del puerto serie, liberando los recursos de software y hardware alocados por la función OpenSerialPort .
SendSerialMessage	Envía un mensaje por el puerto serie.
ResendSerialMessage	Re-envía el último mensaje por el puerto serie.

	TÍTULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 5

Confidencial

2.1.1 OpenSerialPort

Efectúa la apertura del puerto serie, iniciando de este modo la comunicación entre el microcomputador y la impresora fiscal. Una llamada realizada con éxito a esta función es pre-requisito para todo el funcionamiento de la librería.

Sintaxis:

```
DWORD OpenSerialPort( WORD wCommNumber, DWORD dwBaudRate );
```

Entradas:

wCommNumber	Número del Puerto Serie que será utilizado para la comunicación con la impresora fiscal. Soporta valores de 1 a 4 para PCs con direccionamiento standard de Coms .
dwBaudRate	Velocidad del puerto serie.

Salidas:

No existen.

Retorno:

EFPROT_SUCCESS	Port abierto con éxito y recursos alocados para su utilización.
EFPROT_PORT_ALREADY_OPEN	Apertura del puerto serie ya efectuada.
EFPROT_PORT_IN_USE	Puerto serie en uso por otra aplicación.
EFPROT_INVALID_COM	Puerto serie utilizado para la apertura no existe.
EFPROT_INVALID_BAUD_RATE	Velocidad utilizada para la apertura inválida.
EFPROT_INTERNAL_ERROR	Error Interno de la Librería.

Portabilidad:



☒ Win 32

☒ Dos

☒ Linux

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 6

Confidencial

Declaración en C :



```
typedef unsigned int  dword;  
typedef unsigned short word;
```

```
dword OpenSerialPort( word wCommNumber, dword dwBaudRate ); //Prototipo.
```

```
# define COM1                1  
# define BAUD_RATE          9600
```

```
dword dwReturn; //Declara la variable.  
dwReturn = OpenSerialPort( COM1, BAUD_RATE ); //Abre el COM1 con 9600
```

Declaración en Visual Basic:



```
Private Declare Function OpenSerialPort Lib "EpsonFiscalProtocol.dll" (ByVal iSerialPort As Integer, ByVal iBaudRate As Long) As Long
```

Declaración en Visual C++ / C++ Builder:



```
typedef int (*LPOPENPORT)( WORD, DWORD );  
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
LPOPENPORT pOpenPort = (LPOPENPORT) GetProcAddress( HandleLib, "OpenSerialPort");
```

Cuidado:



Las variables **HandleLib** y **pOpenPort** deben tener sus valores diferentes de nulo para ser utilizadas.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 7

Confidencial

2.1.2 CloseSerialPort

Efectúa el cierre del puerto serie, cerrando de este modo la comunicación entre el microcomputador y la impresora fiscal.

Sintaxis:

```
dword CloseSerialPort( void );
```

Entradas:

No existen.

Salidas:

No existen.

Retorno:

EFPROT_SUCCESS
EFPROT_NOT_OPEN

Puerto Serie cerrado con éxito.
Puerto Serie no abierto por la función
OpenSerialPort.

Portabilidad:



☒ Win 32

☒ Dos

☒ Linux

Declaración en C :



```
typedef unsigned int dword;
```

```
dword CloseSerialPort( void ); //Prototipo.
```

```
dword dwReturn; //Declara variable.
```

```
dwReturn = CloseSerialPort (); // cierra puerto serie
```

Declaración en Visual Basic:



```
Private Declare Function CloseSerialPort Lib "EpsonFiscalProtocol.dll" () As Long
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 8

Confidencial

Declaración en Visual C++ / C++ Builder:



```
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
FARPROC pClosePort = GetProcAddress( HandleLib, "CloseSerialPort");
```

Cuidado:



Las variables **HandleLib** y **pClosePort** deben tener sus valores diferentes de nulo para ser utilizadas.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 9

Confidencial

2.1.3 SendSerialMessage

Envía un mensaje almacenado en el buffer de envío para la impresora fiscal. Esta función será de funcionamiento asíncronico para los sistemas Win 32 y Linux, y retornará el control al aplicativo mientras el proceso de impresión es ejecutado en Background. Para el ambiente Dos esta función tiene funcionamiento sincrónico y solamente retornará el control al aplicativo después de la liberación de la impresora.

Sintaxis:

```
dword SendSerialMessage( void );
```

Entradas:

No existen.

Salidas:

No existen.

Retorno:

EFPROT_SUCCESS	Thread de envío de datos iniciado con éxito.
EFPROT_INTERNAL_ERROR	Error interno de la librería.

Portabilidad:



☒ Win 32

☒ Dos

☒ Linux

Declaración en C:



```
typedef unsigned int dword;
```

```
dword SendSerialMessage( void ); //Prototipo.
```

```
dword dwReturn; //Declara la variable.
```

```
dwReturn = SendSerialMessage(); // Envía un mensaje.
```

Declaración en Visual Basic:



```
Private Declare Function SendSerialMessage Lib "EpsonFiscalProtocol.dll" () As Long
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 10

Confidencial

Declaración en Visual C++ / C++ Builder:



```
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
FARPROC pSendMessage = GetProcAddress( HandleLib, "SendSerialMessage");
```

Cuidado:



Las variables **HandleLib** y **pSendMessage** deben tener sus valores diferentes de nulo para ser utilizadas.

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 11

Confidencial

2.1.4 ResendSerialMessage

Reenvía el último mensaje enviado a la impresora fiscal. Esta función será de funcionamiento asíncrono para los sistemas Win 32 y Linux, y retornará el control al aplicativo mientras el proceso de impresión es ejecutado en Background. Para el ambiente Dos esta función tiene funcionamiento síncrono y solamente retornará el control al aplicativo después de la liberación de la impresora.

Sintaxis:

```
dword ResendSerialMessage( void );
```

Entradas:

No existen.

Salidas:

No existen.

Retorno:

EFPROT_SUCCESS	Thread de envío de datos iniciada con éxito.
EFPROT_INTERNAL_ERROR	Error interno de la librería.

Portabilidad:



☒ Win 32

☒ Dos

☒ Linux

Declaración en C:



```
typedef unsigned int dword;
```

```
dword ResendSerialMessage( void ); //Prototipo.
```

```
dword dwReturn; //Declara la variable.
```

```
dwReturn = ResendSerialMessage (); // Reenvía el último mensaje
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 12

Confidencial

Declaración en Visual Basic:



Private Declare Function ResendSerialMessage Lib "EpsonFiscalProtocol.dll" () As Long

Declaración en Visual C++ / C++ Builder:



HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");
FARPROC pResendMessage = GetProcAddress(HandleLib, "ResendSerialMessage");

Cuidado:



Las variables **HandleLib** y **pResendMessage** deben tener valores diferentes de nulo para ser utilizadas.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 13

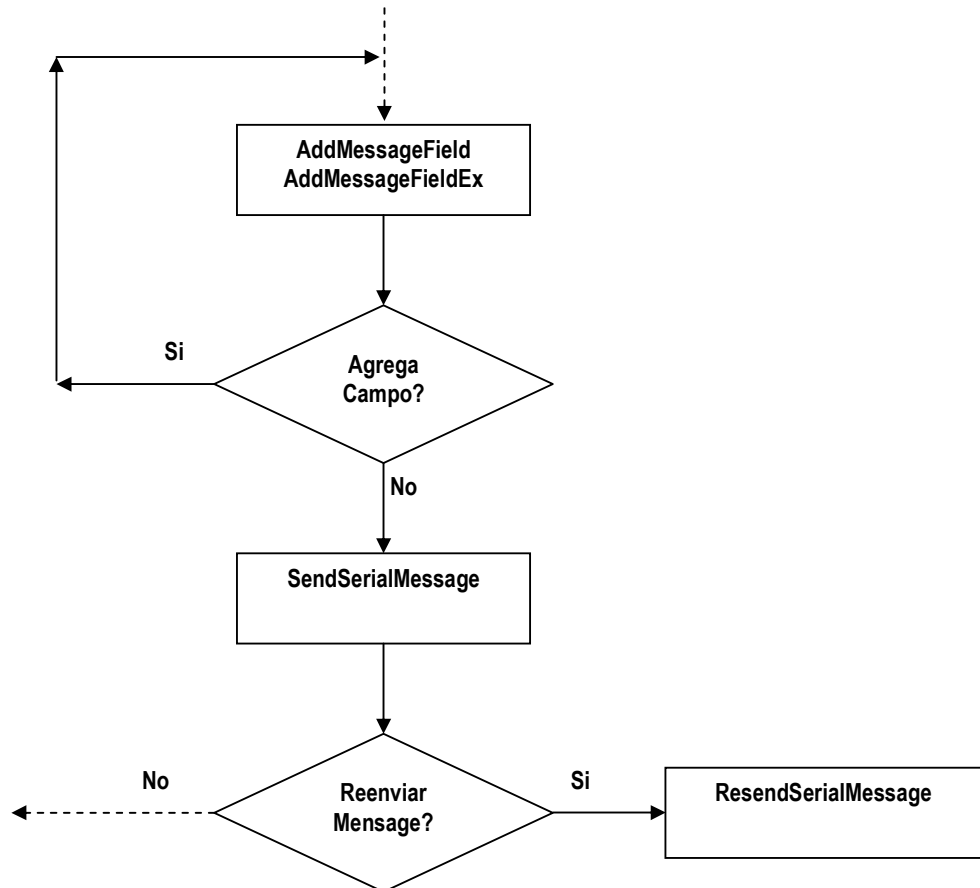
2.2 Funciones de Manejo del Buffer de Salida

Las funciones descritas en esta sección son utilizadas para agregar datos al buffer de salida, que serán enviados por el puerto serial a través de las funciones de envío descritas en la sección anterior. Las funciones son la siguientes:

Función	Descripción
AddMessageField	Agrega datos binarios en el buffer de salida.
AddMessageFieldEx	Agrega datos, en el formato especificado, al buffer de salida.
PurgeMessage	Limpia el buffer de salida.



El siguiente flujograma demuestra la utilización de estas funciones en conjunto con las funciones **SendSerialMessage** y **ResendSerialMessage** para el envío de datos a la Impresora Fiscal.



EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 14

Confidencial

2.2.1 AddMessageField

Agrega datos al buffer de salida, en formato binario. El separador de campos también es agregado cuando es necesario.

Sintaxis:

```
dword AddMessageField( byte * szField, dword dwFieldLength);
```

Entradas:

szField	Campo que será agregado al buffer de salida, que será enviado a la impresora fiscal.
dwFieldLength	Tamaño del campo szField ..

Saídas:

No existen.

Retorno:

EFPROT_SUCCESS	Datos agregados al buffer de salida con éxito.
EFPROT_BUFFER_OVERFLOW	Límite del buffer de salida alcanzado. Dato no agregado.
EFPROT_FIELD_INVALID	Formato inválido de los datos en szField .
EFPROT_INVALID_PARAM	Parámetro szField nulo.

Portabilidad:



☒ Win 32 ☒ Dos ☒ Linux

Declaración en C:



```
typedef unsigned int  dword;  
typedef unsigned char byte;
```

```
dword AddMessageField( byte * szField, dword dwFieldLength); //Prototipo.
```

```
dword dwReturn; //Declara la variable.  
byte bField[] = { 0x08, 0x02 }; //Campo com tamaño 2.
```

```
dwReturn = AddMessageField( bField, 2 ); // Agrega el Campo en el Buffer de salida.
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 15

Confidencial

Declaración en Visual Basic:



Private Declare Function AddMessageField Lib "EpsonFiscalProtocol.dll" (ByVal szField As String, ByVal IFieldLength As Long) As Long

Declaración en Visual C++ / C++ Builder:



```
typedef int ( *LPAMSG)( BYTE *, DWORD );  
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
LPAMSG pAddMsgField = (LPAMSG) GetProcAddress( HandleLib, "AddMessageField");
```

Cuidados:



Las variables **HandleLib** y **pAddMsgField** deben tener sus valores diferentes de nulo para ser utilizadas.

Esta función interpretará los datos pasados en el campo **szField** como binarios, es decir, utilizará el código ASCII correspondiente de los caracteres. Por ejemplo, si el argumento **szField** es igual a "0802", la función agregará en el buffer los datos 30h, 38h, 30h e 32h y no 08h e 02h.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 16

Confidencial

Código Ejemplo en Visual C++:

```
typedef int ( * LPAMSG )( BYTE *,DWORD );
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");

void AddMessageField( void )
{
    LPAMSG      fAddMessageField;
    unsigned char szDatos[2];

    if( HandleLib == NULL )
    {
        /* Error Abriendo la DLL. Efectuar el tratamiento y retornar */
        return;
    }

    fAddMessageField = (LPAMSG) GetProcAddress( HandleLib,"AddMessageField");

    if( fAddMessageField == NULL )
    {
        /* Error Leyendo función de la DLL. Efectuar el tratamiento y retornar */
        return;
    }

    /* Se agrega el dato 0802h al buffer de salida de la impresora */
    szDatos[0] = 0x08;
    szDatos[1] = 0x02;
    if( fAddMessageField( szDatos, 0x02 ) != 0 )
    {
        /* Error agregando datos. Efectuar el tratamiento y retornar */
        return;
    }
}
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 17

Confidencial

Código Ejemplo en C:

```
typedef unsigned int  dword;
typedef unsigned char bool;
typedef unsigned char byte;

//prototipo
dword AddMessageField( byte *uchField, dword iFieldLength );

int main( void )
{

//Declaración de las variables
dword dwReturn;
byte bField[] = { 0x08, 0x02 }; //Inicializa bField

//Agrega un mensaje al buffer de salida
dwReturn = AddMessageField( bField, 2 );

//Testea dwReturn
if( dwReturn != 0 )
{
    //Error al agregar. Efectuar el tratamiento y retornar.
    return;
}

}
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 18

Confidencial

2.2.2 AddMessageFieldEx

Agrega datos al buffer de salida, en el formato especificado por el usuario.

Sintaxis:

```
dword AddMessageFieldEx( byte * szField, dword dwLength, dword dwType, bool bSeparator );
```

Entradas:

szField	Campo que será agregado al buffer de salida, que será enviado a la impresora fiscal. Este campo debe estar en formato binario.
dwLength	Tamaño del campo szField .
dwType	Tipo de campo a ser agregado. Puede ser EFPROT_TYPE_BINARY o EFPROT_TYPE_ASCII .
bSeparator	Indica si un separador será o no agregado al final del campo.

Salidas:

No existen.

Retorno:

EFPROT_SUCCESS	Datos agregados al buffer de salida con éxito.
EFPROT_BUFFER_OVERFLOW	Límite del buffer de salida alcanzado. Dato no agregado.
EFPROT_FIELD_INVALID	Formato de los datos en szField inválido.
EFPROT_FIELD_INVALID_TYPE	Campo en dwType inválido.
EFPROT_INVALID_PARAM	Parámetro szField nulo.

Portabilidad:

☒ Win 32

☒ Dos

☒ Linux

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 19

Confidencial

Declaración en C:



```
typedef unsigned int  dword;  
typedef unsigned char bool;  
typedef unsigned char byte;
```

//Prototipo.

```
dword AddMessageFieldEx( byte * szField, dword dwLength, dword dwType, bool bSeparator  
);
```

```
# define TRUE          1  
# define FALSE        0  
# define EFPROT_TYPE_CHAR 1
```

```
dword dwReturn; //Declara la variable.  
byte bField[] = "0800"; //Campo con tamaño 4.
```

```
dwReturn = AddMessageFieldEx( bField, 4, EFPROT_TYPE_CHAR, TRUE ); // Agrega el  
                           campo al Buffer de salida.
```

Declaración en Visual Basic:



```
Private Declare Function AddMessageFieldEx Lib "EpsonFiscalProtocol.dll" (ByVal szField As  
String, ByVal lFieldLength As Long, ByVal lFieldType As Long, ByVal bSeparator As Boolean)  
As Long
```

Declaración en Visual C++ / C++ Builder:



```
typedef int ( *LPAMSGEX )( BYTE *, DWORD );  
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
LPAMSGEX pAddMsgFieldEx = (LPAMSG) GetProcAddress( HandleLib,  
"AddMessageFieldEx");
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 20

Confidencial

Cuidados:



Las variables **HandleLib** y **pAddMsgFieldEx** deben tener valores diferentes a nulo para ser utilizadas.

El comportamiento de esta función difiere de la anterior en el siguiente ejemplo. Caso que se quiera agregar un dato 0802h, se puede llamar directamente a la función **AddMessageFieldEx** con un parámetro **szField** igual a "0802" o un parámetro **dwType** igual a **EFPROT_TYPE_BINARY**. De este modo, la librería convertirá este valor a **08h** y **02h**. Si el parámetro **dwType** es igual a **EFPROT_TYPE_ASCII**, esta función se comportará de modo idéntico a **AddMessageField**.

Para agregar datos en formato binario, se debe tener en mente que el tamaño de estos datos deben ser múltiplos de dos, pues, caso contrario, la conversión a el respectivo valor en hexadecimal no puede ser realizada.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 21

Confidencial

Código Ejemplo en Visual C++:

```
typedef int ( * LPAMSGEX )( BYTE *,DWORD, DWORD, BOOL );
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");

void AddMessageField( void )
{
    LPAMSGEX    fAddMessageFieldEx;
    unsigned char szDatos[5];

    if( HandleLib == NULL )
    {
        /* Error de apertura de DLL. Efectuar el tratamiento y retornar */
        return;
    }

    fAddMessageFieldEx = (LPAMSGEX) GetProcAddress( HandleLib,"AddMessageFieldEx");

    if( fAddMessageFieldEx == NULL )
    {
        /* Error leyendo función de la DLL. Efectuar el tratamiento y retornar */
        return;
    }

    /* Vamos a agregar el dato 0802h en el buffer de salida de la impresora */
    strcpy( (char *) szDatos, "0802");
    if(fAddMessageFieldEx(szDatos,strlen(szDatos),EFPROT_TYPE_BINARY,true ) != 0 )
    {
        /* Error agregando datos. Efectuar el tratamiento y retornar */
        return;
    }
}
```

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 22

Código Ejemplo en C:

```
typedef unsigned int  dword;
typedef unsigned char bool;
typedef unsigned char byte;

dword AddMessageFieldEx( byte * szField, dword dwLength, dword dwType, bool bSeparator );
//Prototipo.

#define FALSE          0
#define TRUE           1
#define EFPROT_TYPE_CHAR 1

int main( void )
{
    //Declaración de variables
    dword dwReturn;
    byte bField[] = "0802"; //Inicializa szField

    //Agrega un mensaje al buffer de salida con el tipo definido y con separador.
    dwReturn = AddMessageFieldEx( bField, 4, EFPROT_TYPE_CHAR, TRUE );

    //Testea el retorno
    if( dwReturn != 0 )
    {
        //Error en el agregado. Efectuar el tratamiento y retornar.
        return;
    }
}
```

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 23

Confidencial

2.2.3 PurgeMessage

Remueve todos los datos del buffer de salida para que un nuevo mensaje sea agregado al mismo.

Sintaxis:

```
void PurgeMessage( void );
```

Entradas:

No existen.

Salidas:

No existen.

Retorno:

No existen.

Portabilidad:☒ Win 32☒ Dos☒ Linux

Declaración en C:

```
void PurgeMessage( void ); //prototipo
```

```
PurgeMessage();
```

Declaración en Visual Basic:

```
Private Declare Sub PurgeMessage Lib "EpsonFiscalProtocol.dll" ()
```

Declaración en Visual C++ / C++ Builder:

```
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
FARPROC pAddMsgField = (FARPROC) GetProcAddress( HandleLib, "PurgeMessage");
```


	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 24

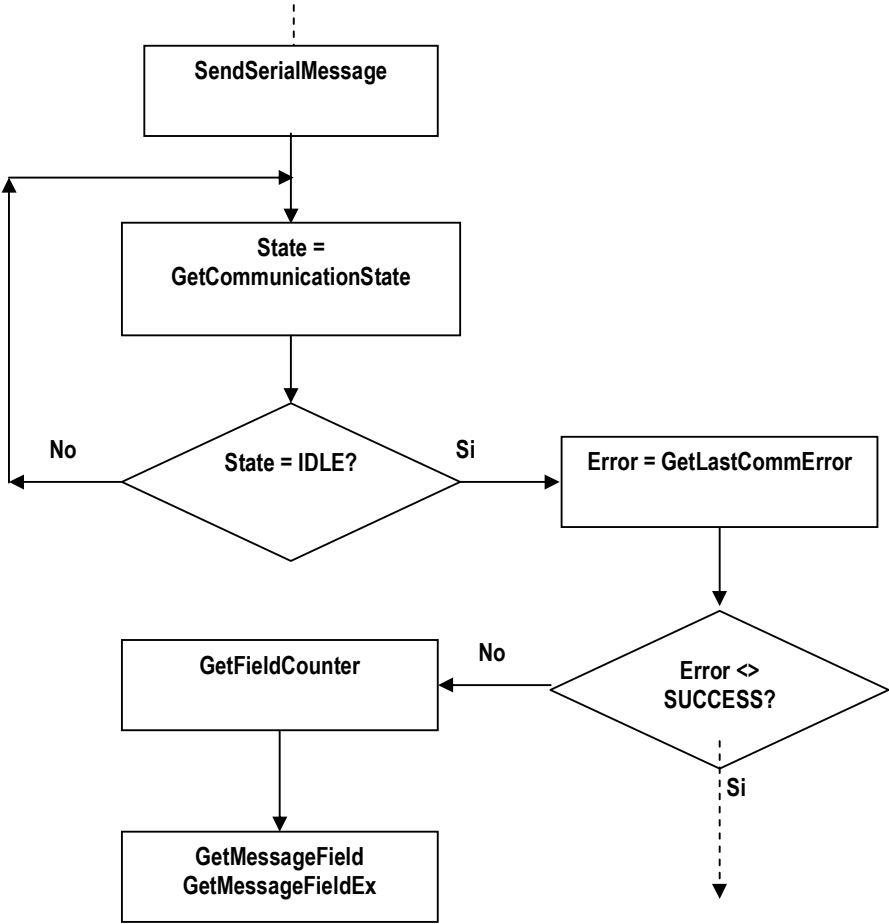
2.3 Funciones de Manejo del Buffer de Entrada

Las funciones descritas en esta sección son utilizadas para capturar datos del buffer de entrada, que serán recibidos por el puerto serial. Estas funciones son:


Función	Descripción
GetMessageField	Retorna un campo del buffer de respuesta en formato binario.
GetMessageFieldEx	Retorna un campo del buffer de respuesta en el formato solicitado.
GetFieldCounter	Retorna el número de campos continuos en el buffer de respuesta.

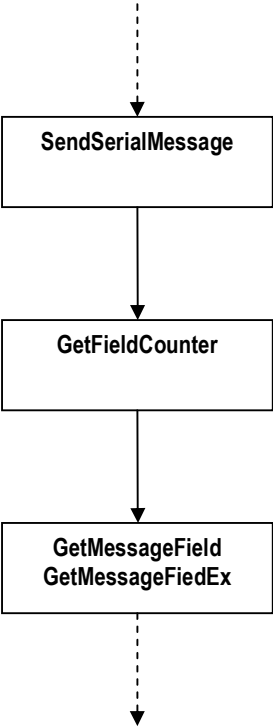
2.3.1 Para sistemas Win 32 e Linux

 El siguiente flujograma demuestra un procedimiento básico de utilización de estas rutinas.



2.3.2 Para sistema Dos

 El siguiente flujograma demuestra un procedimiento básico de utilización de estas rutinas.



Confidencial

2.3.3 GetMessageField

Captura un campo específico del buffer de respuesta de la impresora fiscal. Todos los datos retornados por esta función están en formato binario.

Sintaxis:

```
DWORD GetMessageField( byte * szField, DWORD * dwFieldLength, DWORD dwAnswerField);
```

Entradas:

dwAnswerField	Posición del campo a ser recuperado del buffer de respuesta de la impresora fiscal.
----------------------	---

Salidas:

szField	Campo solicitado esta vacío en caso de error.
dwFieldLength	Tamaño en bytes del campo szField .

Retorno:

EFPROT_SUCCESS	Datos capturados del buffer de entrada con éxito.
EFPROT_ASWR_INVALID	Campo de respuesta solicitado inválido.
EFPROT_ASWR_FIELD_INEXISTENT	Campo de respuesta solicitado no existe.
EFPROT_FIELD_INVALID_TYPE	Tipo de campo de respuesta inválido.
EFPROT_INTERNAL_ERROR	Error interno de la librería.
EFPROT_INVALID_PARAM	Parámetro szField y/o dwFieldLength nulo.

Portabilidad:

☒ Win 32☒ Dos☒ Linux

EPSON	TÍTULO LIBRERÍA DE COMUNICACIÓN BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISIÓN HOJA 1.4	NRO.	
			PROXIMA -	HOJA 27

Confidencial

Declaración en C:



```
typedef unsigned int  dword;  
typedef unsigned char byte;
```

//Prototipo.

```
dword GetMessageField( byte * szField, dword * dwFieldLength, dword dwAnswerField);
```

```
# define EFPROT_PRT_ANSWER          1
```

```
dword dwReturn, dwLen; //Declara la variable.  
byte bAnswer[5];
```

// Captura Campo de repuesta

```
dwReturn = GetMessageField ( bAnswer, &dwLen, EFPROT_PRT_ANSWER );
```

Declaración en Visual Basic:



```
Private Declare Function GetMessageField Lib "EpsonFiscalProtocol.dll" (ByVal szField As  
String, ByRef lFieldLength As Long, ByVal lAnswerField As Long) As Long
```

Declaración en Visual C++ / C++ Builder:



```
typedef int ( *LPGMSG)( BYTE *, DWORD *, DWORD );  
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
LPGMSG pGetMsgField = (LPGMSG) GetProcAddress( HandleLib, "GetMessageField");
```

Cuidados:



La variableis **HandleLib** y **pGetMsgField** deben tener sus valores diferentes de nulo para ser utilizadas.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 28

Confidencial

Código Ejemplo en Visual C++:

```
typedef int ( * LPGMSG )( BYTE *,DWORD *, DWORD );
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");

void GetMessageField( void )
{
    LPGMSG      fGetMessageField;
    unsigned char szDatos[2];
    int         iFieldLength;

    if( HandleLib == NULL )
    {
        /* Error abriendo DLL. Efectuar el tratamiento y retornar */
        return;
    }

    fGetMessageField = (LPGMSG) GetProcAddress( HandleLib,"GetMessageField");

    if( fGetMessageField == NULL )
    {
        /* Error leyendo función de la DLL. Efectuar el tratamiento y retornar */
        return;
    }

    /* Se lee el campo "Printer Answer" del buffer de respuesta de la impresora */
    if( fGetMessageField( szDatos, &iFieldLength, EFPROT_PRT_ANSWER ) != 0 )
    {
        /* Error recuperando campo. Efectuar el tratamiento y retornar */
        return;
    }

    /* Campo retornado con éxito. Efectuar el tratamiento y retornar */
    ...
}
```

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 29

Confidencial

Código Ejemplo en C:

```
typedef unsigned int  dword;
typedef unsigned char bool;
typedef unsigned char byte;

//prototipo
dword GetMessageField( byte * szField, dword * dwFieldLength, dword dwAnswerField);

# define EFPROT_PRT_ANSWER          1

int main( void )
{
    //Declaración de variables
    dword dwReturn, dwLen;
    byte bField[5];

    //Obtener el contenido del campo especificado - Status de la impresora
    dwReturn = GetMessageField( bAnswer, 4, &dwLen, EFPROT_PRT_ANSWER );

    //Testear el retorno
    if( dwReturn != 0 )
    {
        //Error al agregar. Efectuar el tratamiento y retornar.
        return;
    }
}
```

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 30

Confidencial

2.3.4 GetMessageFieldEx

Captura un campo específico del buffer de respuesta de la impresora fiscal. El formato del campo de respuesta puede ser escogido a través del parámetro **dwFieldType**, de acuerdo con a tabla de formatos disponibles.

Sintaxis:

```
dword GetMessageFieldEx( void * vField, dword * dwLength, dword dField, dword dwType);
```

Entradas:

dwField	Posición del campo a ser recuperado del buffer de respuesta de la impresora fiscal.
dwType	Tipo de formato a ser aplicado el la variable apuntada por vField .

Salidas:

vField	Campo solicitado en vacio en caso de error.
dwLength	Tamaño en bytes del campo szField .

Retorno:

EFPROT_SUCCESS	Dados capturados del buffer de entrada con éxito.
EFPROT_ASWR_INVALID	Campo de respuesta solicitado inválido.
EFPROT_ASWR_FIELD_INEXISTENT	Campo de respuesta solicitado no existe.
EFPROT_FIELD_INVALID_TYPE	Tipo de campo de respuesta inválido.
EFPROT_INTERNAL_ERROR	Error interno de la librería.
EFPROT_INVALID_PARAM	Parámetro szField y/o dwFieldLength nulo.

Portabilidad:



☒ Win 32

☒ Dos

☒ Linux

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 31

Confidencial

Declaración en C:



```
typedef unsigned int  dword;  
typedef unsigned char byte;
```

//Prototipo.

```
dword GetMessageFieldEx( void * vField, dword * dwLength, dword dField, dword dwType);
```

```
# define EFPROT_PRT_ANSWER          1  
# define EFPROT_TYPE_STRING        8
```

```
dword dwReturn, dwLen; //Declara la variable.  
byte bAnswer[ 5 ]; .
```

// obtiene el contenido del Campo del Buffer de salida.

```
dwReturn = GetMessageFieldEx( bAnswer, &dwLen, EFPROT_PRT_ANSWER,  
EFPROT_TYPE_STRING );
```

Declaración en Visual Basic:



```
Private Declare Function GetMessageFieldEx Lib "EpsonFiscalProtocol.dll" (ByRef iField As  
Long, ByRef iFieldLength As Long, ByVal lAnswerField As Long, ByVal iType As Long ) As  
Long
```

Declaración en Visual C++ / C++ Builder:



```
typedef int ( *LPGMSGEX)( VOID *, DWORD *, DWORD, DWORD );  
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
LPGMSGEX pGetMsgFieldEx = (LPGMSGEX) GetProcAddress( HandleLib,  
"GetMessageFieldEx");
```

Cuidados:



Las variables **HandleLib** y **pGetMsgFieldEx** deben tener sus valores diferentes de nulo para ser utilizadas.

Debido a la variedad de tipos que esta función puede retornar se debe, en Visual Basic, declararla diversas veces, apenas cambiando el tipo del primer parámetro.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 32

Código Ejemplo en Visual C++:

```
typedef int ( * LPGMSGEX )( VOID *,DWORD *, DWORD, DWORD );
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");

void GetMessageFieldEx( void )
{
    LPGMSGEX    fGetMessageFieldEx;
    unsigned short sDatos;
    int          iFieldLength;

    if( HandleLib == NULL )
    {
        /* Error abriendo la DLL. Efectuar el tratamiento y retornar */
        return;
    }

    fGetMessageFieldEx = (LPGMSGEX) GetProcAddress( HandleLib,"GetMessageFieldEx");

    if( fGetMessageFieldEx == NULL )
    {
        /* Error leyendo función de la DLL. Efectuar el tratamiento y retornar */
        return;
    }

    /* Se lee el campo "Printer Answer" del buffer de respuesta de la impresora */
    if( fGetMessageFieldEx( sDatos, &iFieldLength, EFPROT_PRT_ANSWER,
                           EFPROT_TYPE_USHORT ) != 0 )
    {
        /* Error recuperando campo. Efectuar el tratamiento y retornar */
        return;
    }

    /* Campo retornado con éxito. Efectuar el tratamiento y retornar */
    ...
}
```

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 33

Código Ejemplo en C:

```
typedef unsigned int  dword;
typedef unsigned char byte;

//prototipo
dword GetMessageFieldEx( void * vField, dword * dwLength, dword dField, dword dwType);

#define EFPROT_PRT_ANSWER      1
#define EFPROT_TYPE_STRING    8

int main( void )
{
    //Declaración de variables
    dword dwReturn, dwLen;
    byte bAnswer[5];

    //Obtener el contenido del campo especificado - Status de la impresora
    dwReturn = GetMessageFieldEx( bAnswer, &dwLen, EFPROT_PRT_ANSWER,
    EFPROT_TYPE_STRING );

    //Testear el retorno
    if( dwReturn != 0 )
    {
        //Error al agregar. Efectuar el tratamiento y retornar.
        return;
    }
}
```

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 34

Confidencial

2.3.5 GetFieldCounter

Esta función retorna un número total de campos recibidos de la impresora fiscal.

Sintaxis:

```
dword GetMessageField( void );
```

Entradas:

No existen.

Salidas:

No existen.

Retorno:

Número de campos en el buffer de entrada.

Portabilidad:



☒ Win 32

☒ Dos

☒ Linux

Declaración en C:



```
typedef unsigned int dword;
```

```
dword GetMessageField( void ); //prototipo
```

```
dword dwReturn;
```

```
dwReturn = GetFieldCounter();
```

Declaración en Visual Basic:



```
Private Declare Function GetFieldCounter Lib "EpsonFiscalProtocol.dll" () As Long
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 35

Confidencial

Declaración en Visual C++ / C++ Builder:



```
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
FARPROC pFieldCounter = GetProcAddress( HandleLib, "GetFieldCounter");
```

Cuidados:



Las variables **HandleLib** y **pGetFieldCounter** deben tener valor diferente de nulo para ser utilizadas.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 36

2.4 Funciones de Status

Utilizadas para capturar el status actual de la comunicación al igual que para conocer el último error ocurrido. No se debe enviar o recibir datos antes de efectuar una llamada a estas funciones para verificar si todo está ok. Las funciones son las siguientes:

Función	Descripción
GetCommunicationState	Retorna el Status actual de las funciones de comunicación.
GetLastCommError	Retorna el Error ocurrido en la última operación de comunicación serial.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 37

Confidencial

2.4.1 GetCommunicationState

Retorna el estado actual de la comunicación serial.

Sintaxis:

```
dword GetCommunicationState( void );
```

Entradas:

No existen.

Salidas:

No existen.

Retorno:

EFPROT_CLOSED	Puerto serie de comunicación cerrado.
EFPROT_IDLE	Librería en estado Idle. Nuevos comandos pueden ser ejecutados.
EFPROT_BUSY	Librería efectuando tareas en background.

Portabilidad:



☒ Win 32

☒ Linux

Declaración en C:



```
typedef unsigned int dword;  
  
dword GetCommunicationState( void ); //prototipo  
  
dword dwReturn;  
  
dwReturn = GetCommunicationState ();
```

Declaración en Visual Basic:



Private Declare Function GetCommunicationState Lib "EpsonFiscalProtocol.dll" () As Long

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 38

Confidencial

Declaración en Visual C++ / C++ Builder:



```
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
FARPROC pCommState = GetProcAddress( HandleLib, "GetCommunicationState");
```

Cuidados:



Las variables **HandleLib** y **pGetCommStatus** deben tener sus valores diferentes de nulo para ser utilizadas.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 39

Confidencial

2.4.2 GetLastCommError

Retorna el último error ocurrido en la comunicación con la impresora fiscal.

Sintaxis:

```
dword GetLastCommError( void );
```

Entradas:

No existen.

Salidas:

No existen.

Retorno:

EFPROT_SUCCESS	Operación completada con éxito.
EFPROT_TIMEOUT_ERROR	Time-out de comunicación.
EFPROT_SEND_ERROR	Error de envío de datos.
EFPROT_OFF_OR_DISCONNECTED	Impresora off-line o desconectada.

Portabilidad:



☒ Win 32

☒ Linux

Declaración en C:



```
typedef unsigned int dword;
```

```
dword GetLastCommError( void ); //prototipo
```

```
dword dwReturn;
```

```
dwReturn = GetLastCommError();
```

Declaración en Visual Basic:



```
Private Declare Function GetLastCommError Lib "EpsonFiscalProtocol.dll" () As Long
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 40

Confidencial

Declaración en Visual C++ / C++ Builder:



```
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
FARPROC pCommError = GetProcAddress( HandleLib, "GetLastCommError");
```

Cuidados:



Las variables **HandleLib** y **pCommError** deben tener valores diferentes de nulo para ser utilizadas.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 41

2.5 Funciones Informativas

Utilizadas para obtener información de la librería.

Función	Descripción
GetApiVersion	Retorna la versión actual de la librería.
GetSentFrame	Retorna lo último enviado a la impresora.
GetReceivedFrame	Retorna lo último recibido del buffer de la impresora.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 42

Confidencial

2.5.1 GetApiVersion

Retorna la versión de la API.

Sintaxis:

```
void GetApiVersion(Byte * szVersion);
```

Entradas:

No existen.

Saídas:

No existen.

Retorno:

Versión de la API en formato XX.YY.

Portabilidad:



☒ Win 32

☒ Dos

☒ Linux

Declaración en C:



```
typedef unsigned char byte;
```

```
void GetApiVersion( byte * szVersion ); //prototipo
```

Declaración en Visual Basic:



```
Private Declare Sub GetApiVersion Lib "EpsonFiscalProtocol.dll" (szVersion As String)
```

Declaración en Visual C++ / C++ Builder:



```
typedef void (*FARPROC)( BYTE *);  
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
FARPROC pApiVersion = (FARPROC) GetProcAddress( HandleLib, "GetApiVersion");
```

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 43

Cuidados:



Las variables **HandleLib** y **pApiVersion** deben tener valores diferentes de nulo para ser utilizadas.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 44

Confidencial

2.5.2 GetSentFrame

Captura lo último enviado al buffer de la impresora fiscal.

Sintaxis:

```
DWORD GetSentFrame( byte * szBuffer, DWORD * dwBufferLength );
```

Entradas:

No existen.

Salidas:

szBuffer	Último enviado al buffer de la impresora fiscal.
dwBufferLength	Tamaño en bytes del buffer szBuffer .

Retorno:

EFPROT_SUCCESS	Datos capturados del buffer de entrada con éxito.
EFPROT_INTERNAL_ERROR	Error interno de la librería.

Portabilidad:



☒ Win 32

☒ Dos

☒ Linux

Declaración en C:



```
typedef unsigned int  DWORD;  
typedef unsigned char byte;
```

```
DWORD GetSentFrame( byte * szBuffer, DWORD * dwBufferLength ); //Prototipo.
```

```
DWORD dwReturn, dwLen; //Declara variables.  
byte bBuffer[2048];
```

```
dwReturn = GetSentFrame( bBuffer, &dwLen); // captura último buffer enviado para impresora.
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 45

Confidencial

Declaración en Visual Basic:



Private Declare Function GetSentFrame Lib "EpsonFiscalProtocol.dll" (ByVal szBuffer As String, ByRef iBufferLength As Long) As Long

Declaración en Visual C++ / C++ Builder:



```
typedef int ( *LPGSENTF)( BYTE *, DWORD * );  
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
LPGSENTF pGetSentFrame = (LPGSENTF) GetProcAddress( HandleLib, "GetSentBuffer");
```

Cuidados:



Las variables **HandleLib** y **pGetSentFrame** deben tener valores diferentes de nulo para ser utilizadas.

El tamaño de la variable **szBuffer** debe ser grande y suficiente para soportar el retorno de esta función. El tamaño máximo retornado por esta función es de 4102 bytes.

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 46

Confidencial

2.5.3 GetReceivedFrame

Captura lo último recibido del buffer de la impresora fiscal.

Sintaxis:

```
DWORD GetReceivedFrame( byte * szBuffer, DWORD * dwBufferLength );
```

Entradas:

No existen.

Saídas:

szBuffer	Último recibido del buffer de la impresora fiscal.
dwBufferLength	Tamaño en bytes del buffer szBuffer .

Retorno:

EFPROT_SUCCESS	Datos capturados del buffer de entrada con éxito.
EFPROT_INTERNAL_ERROR	Error interno de la librería.

Portabilidad

☒ Win 32

☒ Dos

☒ Linux

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 47

Confidencial

Declaración en C:



```
typedef unsigned int  dword;  
typedef unsigned char byte;
```

//Prototipo.

```
dword GetReceivedFrame( byte * szBuffer, dword * dwBufferLength );
```

dword dwReturn, dwLen; //Declara las variables.

```
byte bBuffer[2048];
```

```
dwReturn = GetReceivedFrame ( bBuffer, &dwLen); // captura último buffer recibido de la  
impresora.
```

Declaración en Visual Basic:



```
Private Declare Function GetReceivedFrame Lib "EpsonFiscalProtocol.dll" (ByVal szBuffer As  
String, ByRef iBufferLength As Long) As Long
```

Declaración en Visual C++ / C++ Builder:



```
typedef int ( *LPGRECVF)( BYTE *, DWORD * );  
HINSTANCE HandleLib = LoadLibrary("EpsonFiscalProtocol.dll");  
LPGRECVF pGetRecvFrame = (LPGRECVF) GetProcAddress( HandleLib,  
GetReceivedBuffer");
```

Cuidados:



Las variables **HandleLib** y **pGetRecvFrame** deben tener valores diferentes de nulo para ser utilizadas.

El tamaño de la variable **szBuffer** debe ser grande o suficiente para soportar el retorno de esta función. El tamaño máximo retornado por esta función es de 4102 bytes

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 48

Capítulo 3 - Guías de como Compilar y Linkear en Linux

Todas las distribuciones de Linux incluyen GCC, un compilador GNU para los lenguajes C, C++ y Objective-C. El compilador GCC ofrece numerosas opciones para compilar y linkear en código C, C++ y Objective-C.

Este documento muestra solamente los recursos mas básicos. Para mayor información sobre el compilador GCC consulte la documentación de su sistema Linux.

3.1 Compilar y Linkear

El compilador GCC reconoce que un código fue escrito en C o C++ a través de la extensión, o sea, si la extensión es “c” minúscula se trata de un código escrito en lenguaje c, en caso que la extensión sea “C” mayúscula o “cpp”, se trata de un código C++. Para esto se utiliza un comando apropiado llamado g++, creado especialmente para códigos escritos en C++ y el mas importante para hacer links. A continuación un código ejemplo.cpp .

```
//ejemplo.cpp
#include <stdio.h>

typedef unsigned int dword;
typedef unsigned short word;

dword OpenSerialPort ( word iSerialPort, dword iBaudRate ); // abre puerto
dword CloseSerialPort( void );//cierra puerto

#define COM1 1
#define BAUD_RATE 9600

int main( void )
{
    dword dwReturn;

    dwReturn = OpenSerialPort( COM1, BAUD_RATE ); //Abre el puerto serie
    printf("Retorno da funcao OpendSerialPort : %d\n", dwReturn );

    dwReturn = CloseSerialPort(); //cierra puerto serie
    printf("Retorno da funcao CloseSerialPort : %d\n", dwReturn);

    return 0;
}
```

EPSON	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 49

3.1.1 Comando del GCC – Librería estática

El resultado del comando será un ejecutable .exe.

Comando:



```
g++ EpsonFiscalProtocol.a -o exes ejemplo.cpp -lpthread
```

Cuidados:



Como la librería fiscal posee funcionamiento multithread es obligatorio incluir **-lpthread**.

3.1.2 Comando del GCC – Librería dinámica

El resultado del comando será un ejecutable .exe menor.

Comando:



```
g++ -L. EpsonFiscalProtocol.so -o exedy ejemplo.cpp -lpthread
```

Cuidados:



Como la librería fiscal posee funcionamiento multithread es obligatorio incluir **-lpthread**.

	TITULO LIBRERIA DE COMUNICACION BAJO NIVEL ESPECIFICACIONES IMPRESORAS FISCALES	REVISION HOJA 1.4	NRO.	
			PROXIMA -	HOJA 50