



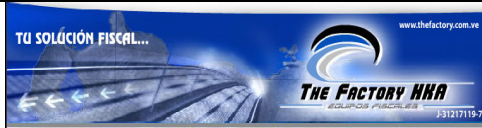
Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

INDICE

1.- Declaración de la Clase Tfhka	1
2.- Métodos de la Clase Tfhka	2
2.1- <i>BOOL OpenFpctrl (string lpPortName)</i>	2
2.2- <i>VOID CloseFpctrl ()</i>	2
2.3- <i>BOOL CheckFprinter ()</i>	2
2.4- <i>BOOL ReadFpStatus ()</i>	2
2.5- <i>BOOL SendCmd (string cmd)</i>	3
2.6- <i>INT SendCmdFile (string file)</i>	3
2.7- <i>BOOL UploadReportCmd (string cmd, string file)</i>	3
2.8- <i>BOOL UploadStatusCmd (string cmd, string file)</i>	4
3.- Atributos y métodos públicos orientados a objetos de la clase Tfhka.	4
3.1- <i>Atributos de objetos.</i>	4
3.2- <i>Métodos de objetos.</i>	7
4.- Anexos	10
Anexo 1: “Información del Status de la Impresora Fiscal”.	10
Anexo 2: “Comandos Para Leer el Estado de la Impresora”.	12
Anexo 3: “Configuración del juego de caracteres en Linux”.	17

Nota 1: Agregue la referencia TFHKADIR.dll en el proyecto o solución de C# que va ha emplear la clase.

Nota 2: Importe el espacio de nombre TFHKADIR en donde se va ha emplear la clase Tfhka. (using TFHKADIR;)



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

1.- DECLARACIÓN DE LA CLASE Tfhka

Una vez agregada la referencia de la clase TFHKADIR.DLL a la Solución o a un espacio de nombre de la misma y la importación respectiva se puede empezar a trabajar con la clase de protocolo directo Tfhka, con sus atributos y métodos públicos.

1.1 Declaración de un objeto tipo Tfhka.

<Tipo de modificador> Tfhka <Nombre del Objeto>

Ejemplos:

```
private    Tfhka  Mi_Enlace;  
  
public    Tfhka  Mi_Enlace2;
```

1.2 Inicialización del objeto Tfhka.

La clase Tfhka tiene un solo constructor que recibe como parámetro un string ó cadena de carácter con el nombre del puerto serial ó terminar serial.

<Nombre del Objeto> = new Tfhka(<Nombre del Puerto serial>)

Ejemplos:

En Windows:

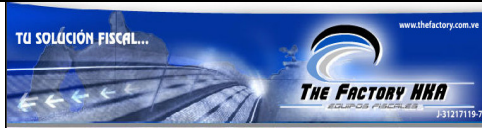
```
Mi_Enlace = new Tfhka ( "COM1" );
```

En Linux:

```
Mi_Enlace = new Tfhka ("/dev/term/a"); // UNIX
```

```
Mi_Enlace = new Tfhka ("/dev/ttyS0"); // Suse
```

Si el parámetro nombre de puerto está correctamente escrito, al efectuar esta inicialización se define el *Comport* de la clase ó string nombre de puerto, se abre si no está en uso y se define el estado del mismo por medio de la variable global booleana *StatusPort*, donde se pone en *true* si el puerto esta abierto y en *false* si el puerto está cerrado o no se pudo abrir.



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

2.- Métodos de la Clase Tfhka

2.1- *BOOL OpenFpctrl (string lpPortName)*

Parámetro: Puerto COM (Ej. “COM1” o “COM2”) ó Terminar “../term/” (“/dev/term/b”)

Método: Apertura del Puerto Serial “COM” ó Terminar “../term/”.

Retorno: Puerto Abierto = **true** ó Falla de Apertura = **false**.

Este método se ejecuta inicialmente en el constructor único de la clase. Se puede volver a ejecutar si se requiere.

Ejemplo:

// Cambio de puerto serial a COM4

```
bool port_status = Mi_Enlace.OpenFpctrl (“COM4”);
```

2.2- *VOID CloseFpctrl ()*

Es el método de la clase que cierra el puerto serial establecido de la misma. Pone a la variable booleana *StatusPort*= **false**.

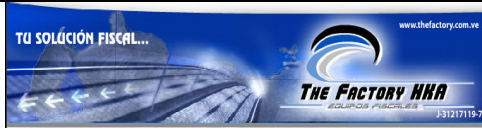
Se ejecuta de esta manera: *Mi_Enlace.CloseFpctrl ();*

2.3- *BOOL CheckFprinter ()*

Método: Verifica si la impresora está Conectada.

Retorno: Impresora Conectada = **true** ó Impresora Desconectada = **false**.

Se ejecuta de esta manera: *Mi_Enlace.CheckFprinter ();*



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

2.4.- BOOL ReadFpStatus ()

Método: Lectura Referente a la Información del Status & Error de la Impresora Fiscal

Retorno: Stand-by = **true** ó Error = **false**.

Se ejecuta de esta manera: `Mi_Enlace.ReadFpStatus ();`

Al llamar a este método se establece el valor de la variable string `Status_Error`. (Ver Anexo 1)

Ejemplo:

El estado de la impresora en modo de espera y sin error sería así:

`Mi_Enlace.Status_Error = "Status: 04 Error: 00";`

2.5- BOOL SendCmd (string cmd)

Parámetro: Comando cmd: Trama de caracteres ASCII. Ejm: "p-1200"

Método: Envía una Línea de Comando a la Impresora Fiscal.

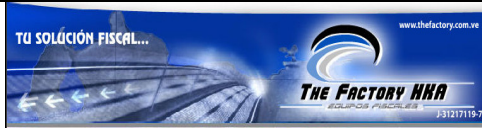
Retorno: Comando Enviado = **true** ó Comando Invalido = **false**.

Ejemplo: Para hacer un reporte Z, *se ejecuta de esta manera:*

`Mi_Enlace.SendCmd ("10Z");`

Al ejecutar este método se establece el estado del envío por medio del string `Envio`.

`Mi_Enlace.Envio = "Status: 00 Error: 00";`



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

2.6- INT SendCmdFile (string file)

Parámetro: Ruta ó Archivo: Ubicación del Archivo.txt ó Archivo.dat.

Método: Envía Comandos a la Impresora Fiscal leído de un archivo de texto plano.

Retorno: Número de comandos válidos.

Ejemplo:

```
Mi_Enlace.SendCmdFile ("C:\MisFiles\Factura1.txt");
```

Al ejecutar este método se realiza envío por medio de los comandos leídos línea por línea del archivo “Factura1.txt”.

NOTA: El archivo de texto plano debe estar guardado con codificación “Unicode” para que se puedan interpretar los caracteres extendidos.

2.7- BOOL UploadReportCmd (string cmd, string file)

Parámetros:

Comando cmd: Comando de Reporte a cargar. Ejm: “U0X”

Ruta ó Archivo: Archivo.txt ó .dat en donde se copiara la información del reporte.

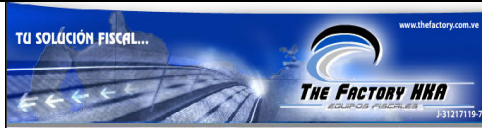
Método: Envía los datos de un Reporte al PC y lo escribe en un archivo de texto plano.

Retorno: Stanby = **true** Error = **false**.

Ejemplo:

```
Mi_Enlace.UploadReportCmd ( "U0Z", "C:\MisFiles\ReporteZ.txt");
```

Al ejecutar este método se copia en el archivo “ReporteZ.txt” la información cargada de un Reporte Z. (Ver Anexo 2)



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

2.8- *BOOL UploadStatusCmd (string cmd, string file)*

Parámetros:

Comando cmd: Estado a subir. Ejm: “S1”

Ruta ó Archivo: Archivo.txt ó .dat en donde se copiara la trama que se subirá.

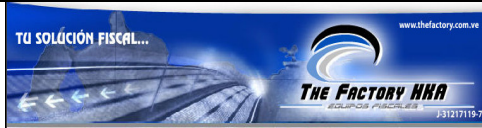
Método: Envía una trama de información al PC del estado leído de la impresora y lo escribe en un archivo de texto plano.

Retorno: Stanby = **true** Error = **false**.

Ejemplo:

```
Mi_Enlace.UploadReportCmd ( “S2” , “C:\MisFiles\Status1.txt”);
```

Al ejecutar este método se lee de la impresora fiscal el estado “S2” y se sube al PC para copiar la trama de información respectiva en el archivo “Status1.txt”. (Ver Anexo 2)



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

3-. Atributos y métodos públicos orientados a objetos de la clase Tfhka.

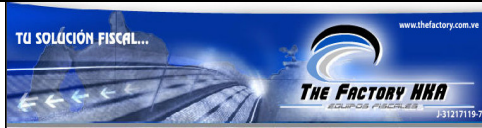
3.1- Atributos de objetos:

- *StatusErrorPrinter* tipo **PrinterStatus**: Representa una estructura que establece y retorna los parámetros de Status y Error de la impresora fiscal cuando se ejecuta el método correspondiente. Sus elementos son los siguientes:

`int` printerStatusCode: Valor entero del Status.
`int` printerErrorCode: Valor entero del Error.
`string` printerStatusdescripcion: Descripción del Status.
`string` printerErrorDescripcion: Descripción del Error.
`bool` ErrorValidity: Indica si Error de la impresora es valido.

- *ReportePC* tipo **ReportData**: Representa una estructura que contiene todos los datos de los reportes X y Z cuando se envían al PC con los métodos correspondientes. Sus elementos son los siguientes:

`int` NumeroDelUltimoReporteZ: Número del último reporte Z.
`int` NumeroDeLaUltimaFactura: Número de la última factura.
`DateTime` Fecha_HoradelaUltimaFactura: Fecha y hora de la última factura.
`double` TotalVentasExento: Total de Ventas en Exento.
`double` TotalImpuestoA: Total del Impuesto A.
`double` TotalImpuestoB: Total del Impuesto B.
`double` TotalImpuestoC: Total del Impuesto C.
`double` TotalVentas: Total de las Ventas.
`double` TotalNotaDeCredito: Total en Notas de Crédito.
`double` TotalNotaDeDebito: Total en Notas de Débito.
`int` NumeroDeLaUltimaNotaDeCredito: Numero de la Ultimo
`int` NumeroDeLaUltimaNotaDeDebito: Monto en devolución Exento.
`int` NumeroDelUltimoDocumentoNoFiscal: Monto en devolución Exento.



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

- *S1Estado* tipo **S1PrinterData**: Representa la estructura de datos del estado S1 que se establece cuando se sube al PC con su método correspondiente. Sus elementos son los siguientes:

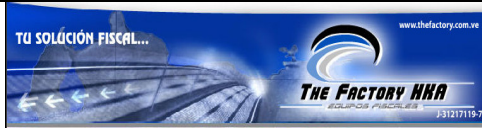
int cashierNumber: Número de cajero asignado.
double totalDailySales: Monto total de ventas diaria.
int lastInvoiceNumber: Número de la última factura emitida.
int quantityOfInvoicesToday: Cantidad de facturas en el día.
int lastNCNumber: Número de la última Nota de Crédito en el día.
int quantityOfNCsToday: Cantidad de Notas de Créditos en el día.
int lastNDNumber: Número de la última Nota de Débito emitida..
int quantityOfNDsToday: Cantidad de Notas de Débito en el día.
int NumberNonFiscalDocuments: Número del Ultimo DNF
int quantityNonFiscalDocuments: Cantidad de DNF en el día.
int dailyClosureCounter: Contador de Cierre Diario Z.
int fiscalReportsCounter: Contador de reporte de Memoria Fiscal.
string RUC: RUC de fiscalización de la impresora fiscal.
string DV: DV de fiscalización de la impresora fiscal.
string registeredMachineNumber: Serial de la impresora fiscal.
DateTime currentPrinterDateTime: Fecha y hora de la impresora fiscal.

- *S2Estado* tipo **S2PrinterData**: Representa la estructura de datos del estado S2 que se establece cuando se sube al PC con su método correspondiente. Sus elementos son los siguientes:

double subTotalBases: Monto sub-total de las Bases Imponibles.
double sudTotalTax: Monto sub-total de IVA.
string dataDummy: Data de relleno. 5
double amountPayable: Monto por pagar.
int numberPaymentsMade: Cantiidad de pagos realizados.
int condition: Condición de transacción.

- *S3Estado* tipo **S3PrinterData**: Representa la estructura de datos del estado S3 que se establece cuando se sube al PC con su método correspondiente. Sus elementos son los siguientes:

int typeTax1: Tipo de tasa 1 (Modo Incluido = 1, Modo Excluido = 2).
double tax1: Valor de la tasa 1 en porcentaje (%).
int typeTax2: Tipo de tasa 2 (Modo Incluido = 1, Modo Excluido = 2).
double tax2: Valor de la tasa 2 en porcentaje (%).
int typeTax3: Tipo de tasa 3 (Modo Incluido = 1, Modo Excluido = 2).



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

double tax3: Valor de la tasa 3 en porcentaje (%).

int[] systemFlags0to49: Lista de valores de los flags del 1 al 49.

- *S4Estado* tipo **S4PrinterData**: Representa la estructura de datos del estado S4 que se establece cuando se sube al PC con su método correspondiente. Sus elementos son los siguientes:

double[] listCumulaMountsMeansPayments: Lista de los montos acumulados de los 16 medios de pagos de la impresora fiscal.

- *S5Estado* tipo **S5PrinterData**: Representa la estructura de datos del estado S5 que se establece cuando se sube al PC con su método correspondiente. Sus elementos son los siguientes:

string RUC: RUC de fiscalización de la impresora.

string DV: DV de fiscalización de la impresora.

string registeredMachineNumber: Serial de la impresora fiscal.

int numberMemoryAudit: Número de memoria de auditoría.

double capacityTotalMemoryAudit: Capacidad de la memoria de auditoría en MB.

double disponyCapacityMemoryAudit: Disponibilidad de la memoria de auditoría en MB.

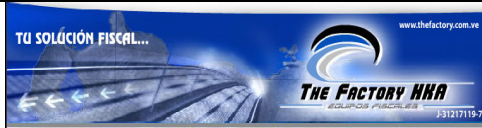
int numberDocumentRegisters: Cantidad de documentos registrados en la memoria de auditoría.

- *ReporteArrayPC* tipo **ReportData[]**: Representa una lista de objetos tipo ReportData que se carga cuando se hace lectura de memoria fiscal con los métodos correspondientes, con una dimensión de un máximo de 2000 elementos posibles. 6

- **PrinterException**: Representa un tipo de excepción que arrojan algunos métodos de creación de objetos de las estructuras anteriormente definidas cuando ocurre un error en la transacción con la impresora fiscal. Está compuesto por los siguientes elementos:

PrinterStatus StatusError: Objeto que contiene información del Status y el Error al momento de generarse la excepción.

Mensaje: Contiene la descripción de la excepción arrojada.



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

3.2- Métodos de Objetos:

3.2.1- PrinterStatus **getPrinterStatus()**.

Método: Establece los datos del objeto *StatusErrorPrinter* de la clase por medio de un llamado interno al método *ReadFpStatus ()*.

Retorno: Un objeto de tipo *PrinterStatus* en la variable global *StatusErrorPrinter* de la clase *Tfhka.cs* con los valores correspondiente a la repuesta generada.

3.2.2- ReportData **getXReport()**.

Método: Sube al PC un reporte X por medio del comando “U0X” estableciendo sus valores de data en el objeto *ReportePC* de la clase *Tfhka.cs*.

Retorno: Un objeto de tipo *ReportData* en la variable global *ReportePC* con toda la información para la carga de un reporte X actual.

Excepción: Arroja la excepción *PrinterExeption*.

3.2.3- ReportData **getZReport()**.

Método: Sube al PC un reporte Z por medio del comando “U0Z” estableciendo sus valores de data en el objeto *ReportePC* de la clase *Tfhka.cs*.

Retorno: Un objeto de tipo *ReportData* en la variable global *ReportePC* con toda la información del último reporte Z emitido.

Excepción: Arroja la excepción *PrinterExeption*.

3.2.4- ReportData[] **getZReport(int StarReportNumber, int EndReportNumber)**.

Parámetros:

- StarReportNumber: Número de reporte Z inicial.
- EndReportNumber: Número de reporte Z final. 7

Método: Realiza una lectura de memoria fiscal por rango de números y asigna los valores de Z(s) en el objeto *ReporteArrayPC*.

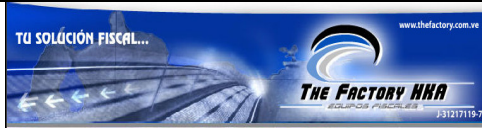
Retorno: Una lista de objetos *ReportData* en la variable *ReporteArrayPC* con la información de los reportes Z comprendidos en el intervalo solicitado.

Excepción: Arroja la excepción *PrinterExeption*.

3.2.5- ReportData[] **getZReport (Date StarDate, Date EndDate)**.

Parámetros:

- StarDate: Fecha inicial.



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

- EndDate: Fecha final.

Método: Realiza una lectura de memoria fiscal por rango de fechas y asigna los valores de Z(s) en el objeto *ReporteArrayPC*.

Retorno: Una lista de objetos ReportData en la variable *ReporteArrayPC* con la información de los reportes Z comprendidos en el intervalo solicitado.

Excepción: Arroja la excepción PrinterExeption.

3.2.6- printZReport(Date StarDate, Date EndDate).

Parámetros:

- StarReportNumber: Número de reporte Z inicial.
- EndReportNumber: Número de reporte Z final.

Método: Imprime una lectura de memoria fiscal por rango de números.

Excepción: Arroja la excepción PrinterExeption.

3.2.7- printZReport (int StarReportNumber, int EndReportNumber).

Parámetros:

- StarDate: Fecha inicial.
- EndDate: Fecha final.

Método: Imprime una lectura de memoria fiscal por rango de fechas.

Excepción: Arroja la excepción PrinterExeption.

3.2.8- printXReport().

Método: Imprime un reporte X.

Excepción: Arroja la excepción PrinterExeption.

8

3.2.9- printZReport().

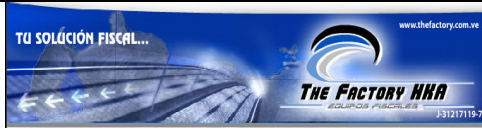
Método: Imprime un reporte Z.

Excepción: Arroja la excepción PrinterExeption.

3.2.10- S1PrinterData getS1PrinterData().

Método: Sube el estado S1 de la Impresora fiscal al PC y estableces los valores respectivos de la data en el objeto *S1Estado* de la clase Tfhka.cs.

Retorno: Un objeto de tipo S1PrinterData en la variable global *S1Estado*.



Librería de Clase “TFHKADIR.DLL” (Clase Protocolo Directo Tfhka)

Excepción: Arroja la excepción PrinterException.

3.2.11- S2PrinterData **getS2PrinterData()**.

Método: Sube el estado S2 de la Impresora fiscal al PC y estableces los valores respectivos de la data en el objeto *S2Estado* de la clase Tfhka.cs.

Retorno: Un objeto de tipo S2PrinterData en la variable global *S2Estado*.

Excepción: Arroja la excepción PrinterException.

3.2.12- S3PrinterData **getS3PrinterData()**.

Método: Sube el estado S3 de la Impresora fiscal al PC y estableces los valores respectivos de la data en el objeto *S3Estado* de la clase Tfhka.cs.

Retorno: Un objeto de tipo S3PrinterData en la variable global *S3Estado*.

Excepción: Arroja la excepción PrinterException.

3.2.13- S4PrinterData **getS4PrinterData()**.

Método: Sube el estado S4 de la Impresora fiscal al PC y estableces los valores respectivos de la data en el objeto *S4Estado* de la clase Tfhka.cs.

Retorno: Un objeto de tipo S4PrinterData en la variable global *S4Estado*.

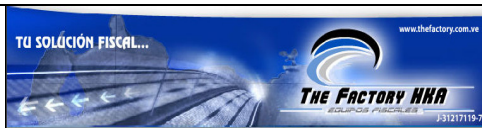
Excepción: Arroja la excepción PrinterException.

3.2.14- S5PrinterData **getS5PrinterData()**.

Método: Sube el estado S5 de la Impresora fiscal al PC y estableces los valores respectivos de la data en el objeto *S5Estado* de la clase Tfhka.cs.

Retorno: Un objeto de tipo S5PrinterData en la variable global *S5Estado*.

Excepción: Arroja la excepción PrinterException.



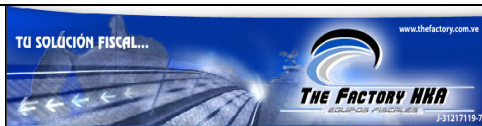
Librería de Clase
“TFHKADIR.DLL”
 (Clase Protocolo Directo Tfhka)

4.- ANEXOS

Anexo 1

“Información del Status de la Impresora Fiscal”

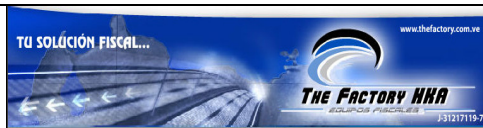
STATUS		
Retorno (Hex)	Retorno (Decimal)	Comentario
0	0	Status Desconocido
1	1	En Modo Prueba y en Espera
2	2	En Modo Prueba y Emisión de Documentos Fiscales
3	3	En Modo Prueba y Emisión de Documentos No Fiscales
4	4	En Modo Fiscal y en Espera
5	5	En Modo Fiscal y Emisión de Documentos Fiscales
6	6	En Modo Fiscal y Emisión de Documentos No Fiscales
7	7	En Modo Fiscal y Cercana Carga Completa De La Memoria Fiscal Y en Espera
8	8	En Modo Fiscal y Cercana Carga Completa De La Memoria Fiscal Y en Emisión de Documentos Fiscales
9	9	En Modo Fiscal y Cercana Carga Completa De La Memoria Fiscal Y en Emisión de Documentos No Fiscales
0A	10	En Modo Fiscal y Carga Completa De La Memoria Fiscal Y en Espera
0B	11	En Modo Fiscal y Carga Completa De La Memoria Fiscal Y en Emisión de Documentos Fiscales
0C	12	En Modo Fiscal y Carga Completa De La Memoria Fiscal Y en Emisión de Documentos No Fiscales



Librería de Clase **“TFHKADIR.DLL”** (Clase Protocolo Directo Tfhka)

“Información del Error de la Impresora Fiscal”

Error			
Retorno (Hex)	Retorno (Decimal)	Comentarios	Valido / Invalido
00	0	No hay Error	VALIDO
01	1	Fin en la Entrega de papel	VALIDO
02	2	Error de índole Mecánico en la entrega de Papel	VALIDO
03	3	Fin en la Entrega de papel y Error Mecánico	VALIDO
50	80	Comando Invalido / Valor Invalido	INVALIDO
54	84	Tasa Invalida	INVALIDO
58	88	No hay Asignadas Directivas	INVALIDO
5C	92	Comando Invalido	INVALIDO
60	96	Error Fiscal	INVALIDO
64	100	Error de la Memoria Fiscal	INVALIDO
6C	108	Memoria Fiscal llena	INVALIDO
70	112	Buffer Completo (Debe enviar el Comando de Reinicio)	INVALIDO
80	128	Error en la Comunicación	INVALIDO
89	137	No Hay Respuesta	INVALIDO
90	144	Error LRC	INVALIDO
91	145	Error Interno API	INVALIDO
99	153	Error en la Apertura del Archivo	INVALIDO



Librería de Clase **“TFHKADIR.DLL”** (Clase Protocolo Directo Tfhka)

Anexo 2.

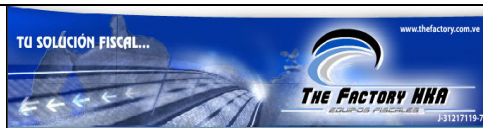
Comandos Para Leer el Estado de la Impresora

Lectura del Estado 1 (S1)

Este comando permite leer desde el host (PC) el estado de la impresora fiscal, referente a parámetros de la impresora como serial, RUC y datos de factura. Es posible ejecutar este comando en cualquier condición.

Detalle de Data de los 145 bytes de Respuesta de la impresora:

Desde	Hasta	Long	Clase	ITEM
1	2	2	ASCII	Comando S1
3	4	2	ASCII	Numero de cajero asignado
5	5	1	0Ah	Separador
6	22	17	ASCII	Total de ventas diarias (14 bytes)
23	23	1	0Ah	Separador
24	31	8	ASCII	Número de la última factura
32	32	1	0Ah	Separador
33	37	5	ASCII	Cantidad de facturas emitidas en el día
38	38	1	0Ah	Separador
39	46	8	ASCII	Número de la última nota de Crédito
47	47	1	0Ah	Separador
48	52	5	ASCII	Cantidad de Notas de Crédito emitidas en el día
53	53	1	0Ah	Separador
54	61	8	ASCII	Número de la última nota de Débito
62	62	1	0Ah	Separador
63	67	5	ASCII	Cantidad de Notas de Débito emitidas en el día
68	68	1	0Ah	Separador
69	76	8	ASCII	Número del último documento no fiscal
77	77	1	0Ah	Separador
78	82	5	ASCII	Cantidad de documentos no fiscales emitidos en el día
83	83	1	0Ah	Separador
84	87	4	ASCII	Contador de cierres diarios
88	88	1	0Ah	Separador
89	92	4	ASCII	Contador de reportes de Memoria Fiscal
93	93	1	0Ah	Separador
94	113	20	ASCII	RUC
114	114	1	0Ah	Separador
115	116	2	ASCII	DV
117	117	1	0Ah	Separador
118	130	13	ASCII	Serial
131	131	1	0Ah	Separador
132	137	6	ASCII	Hora actual en la impresora
138	138	1	0Ah	Separador
139	144	6	ASCII	Fecha actual en la impresora
145	145	1	0Ah	Separador



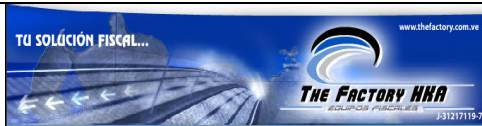
Librería de Clase **“TFHKADIR.DLL”** (Clase Protocolo Directo Tfhka)

Lectura del Estado 2 (S2)

Este comando permite leer desde el host (PC) el estado de la factura en curso en transacción. Si es ejecutado este comando sin una factura en curso, los valores obtenidos serán cero.

Detalle de Data de los 86 bytes de Respuesta de la impresora:

Desde	Hasta	Long	Clase	ITEM
1	2	2	ASCII	Comando S2
3	3	1	20h	Separador (espacio)
4	16	13	ASCII	Subtotal de Bases Imponibles (13 bytes)
17	17	1	0Ah	Separador
18	18	1	20h	Separador (espacio)
19	41	13	ASCII	Subtotal de I.T.B.M
42	42	1	0Ah	Separador
43	43	1	20h	Separador (espacio)
44	56	13	ASCII	Data <u>Dummy</u>
57	57	1	0Ah	Separador
58	63	6	ASCII	Ver Nota*
64	64	1	0Ah	Separador
65	65	1	20h	Separador (espacio)
66	78	13	ASCII	Monto por Pagar
79	79	1	0Ah	Separador
80	83	4	ASCII	Número de Pagos Realizados
84	84	1	0Ah	Separador
85	85	1	ASCII	Condición 0= No transacción 1= En Pago Factura 2= En Nota de Crédito 3= En Nota de Debito
86	86	1	0Ah	Separador



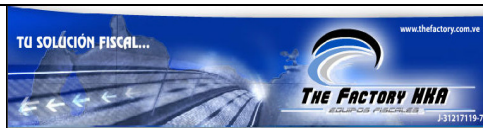
Librería de Clase **“TFHKADIR.DLL”** (Clase Protocolo Directo Tfhka)

Lectura del Estado 3 (S3)

Este comando permite leer desde el host (PC) el estado de la impresora fiscal, referentes a las tasas de impuesto y flags de estado. Es posible ejecutar este comando en cualquier condición.

Detalle de Data de los 121 bytes de Respuesta de la impresora:

Desde	Hasta	SIZE	Clase	ITEM
1	2	2	ASCII	Comando
3	3	1	ASCII	Tipo de tasa 1
4	7	4	ASCII	Valor Tasa 1
8	8	1	0Ah	Separador
9	9	1	ASCII	Tipo de tasa 2
10	13	4	ASCII	Valor Tasa 2
14	14	1	0Ah	Separador
15	15	1	ASCII	Tipo de tasa 3
16	19	4	ASCII	Valor Tasa 3
20	20	1	0Ah	Separador
21	120	100	ASCII	System Flags 1-50 (Cada Flag tiene 2 caracteres)
121	121	1	0Ah	Separador



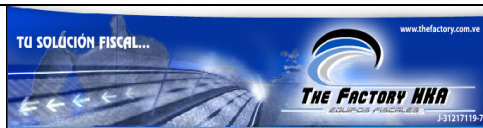
Librería de Clase **“TFHKADIR.DLL”** (Clase Protocolo Directo Tfhka)

Lectura del Estado 4 (S4)

Este comando permite leer desde el host (PC) el estado de la impresora fiscal, referentes a los Medios de Pago. Es posible ejecutar este comando en cualquier condición.

Detalle de Data de los 178 bytes de Respuesta de la impresora:

Desde	Hasta	Long	Clase	ITEM
1	2	2	ASCII	Comando S4
3	12	10	ASCII	Medio de Pago 1
13	13	1	0Ah	Separador
14	23	10	ASCII	Medio de Pago 2
24	24	1	0Ah	Separador
25	34	10	ASCII	Medio de Pago 3
35	35	1	0Ah	Separador
36	45	10	ASCII	Medio de Pago 4
46	46	1	0Ah	Separador
47	56	10	ASCII	Medio de Pago 5
57	57	1	0Ah	Separador
58	67	10	ASCII	Medio de Pago 6
68	68	1	0Ah	Separador
69	78	10	ASCII	Medio de Pago 7
79	79	1	0Ah	Separador
80	89	10	ASCII	Medio de Pago 8
90	90	1	0Ah	Separador
91	100	10	ASCII	Medio de Pago 9
101	101	1	0Ah	Separador
102	111	10	ASCII	Medio de Pago 10
112	112	1	0Ah	Separador
113	122	10	ASCII	Medio de Pago 11
123	123	1	0Ah	Separador
124	133	10	ASCII	Medio de Pago 12
134	134	1	0Ah	Separador
135	144	10	ASCII	Medio de Pago 13
145	145	1	0Ah	Separador
146	155	10	ASCII	Medio de Pago 14
156	156	1	0Ah	Separador
157	166	10	ASCII	Medio de Pago 15
167	167	1	0Ah	Separador
168	177	10	ASCII	Medio de Pago 16
178	178	1	0Ah	Separador



Librería de Clase **“TFHKADIR.DLL”** (Clase Protocolo Directo Tfhka)

Lectura del Estado 5 (S5)

Este comando permite leer desde el host (PC) el estado de la impresora fiscal, referentes a la Capacidad de la Memoria de Auditoria. Es posible ejecutar este comando en cualquier condición.

Detalle de Data de los 64 bytes de Respuesta de la impresora:

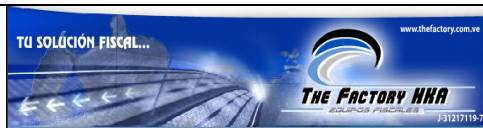
Desde	Hasta	SIZE	Clase	ITEM
1	2	2	ASCII	"S5"
3	22	20	ASCII	RUC
23	23	1	0Ah	Separador 0x0A
24	25	2	ASCII	DV
26	26	1	0Ah	Separador 0x0A
27	39	13	ASCII	Serial
40	40	1	0Ah	Separador 0x0A
41	44	4	ASCII	Número de la memoria de auditoria
45	45	1	0Ah	Separador 0x0A
46	49	4	ASCII	Capacidad de la memoria en MB
50	50	1	0Ah	Separador 0x0A
51	54	4	ASCII	Espacio disponible en la memoria en MB
55	55	1	0Ah	Separador 0x0A
56	63	6	ASCII	Número de documentos registrados
64	64	1	0Ah	Separador 0x0A

Lectura del Estado 6 (S6)

Este comando permite leer desde el host (PC) el estado de la impresora fiscal, referentes a la precencia de papel en las estaciones de impresion. Es posible ejecutar este comando en cualquier condición.

Detalle de Data de los 4 bytes de Respuesta de la impresora:

Desde	Hasta	SIZE	Clase	ITEM
0	1	2	ASCII	"S6"
2	2	1	ASCII	Bit de la estación de impresión de Facturación
3	3	1	ASCII	Bit de la estación de impresión de Slip Endoso
4	4	1	ASCII	Bit de la estación de impresión de Slip Validación



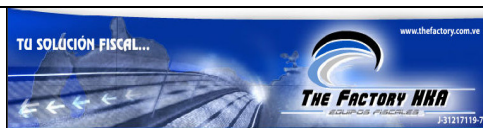
Librería de Clase **“TFHKADIR.DLL”** (Clase Protocolo Directo Tfhka)

Comandos Para Leer Reportes de la Impresora

Lectura del Reporte X (U0X)

Este comando permite leer desde el host (PC) el estado de la impresora fiscal, referente al reporte X. Es posible ejecutar este comando en cualquier condición.

Desde	Hasta	Long	Clase	ITEM
2	5	4	ASCII	Número del Último Reporte Z emitido
6	6	1	0Ah	Separador 0x0A
7	12	6	ASCII	Fecha del último Reporte Z emitido
13	13	1	0Ah	Separador 0x0A
14	17	4	ASCII	Hora del último Reporte Z emitido
18	18	1	0Ah	Separador 0x0A
19	28	10	ASCII	Acumulado Exento
29	29	1	0Ah	Separador 0x0A
30	39	10	ASCII	Acumulado Impuesto Tasa 1
40	40	1	0Ah	Separador 0x0A
41	50	10	ASCII	Acumulado Impuesto Tasa 2
51	51	1	0Ah	Separador 0x0A
52	61	10	ASCII	Acumulado Impuesto Tasa 3
62	62	1	0Ah	Separador 0x0A
63	72	10	ASCII	Total de ventas (sin impuestos)
73	73	1	0Ah	Separador 0x0A
74	83	10	ASCII	Total de nota de crédito (sin impuestos)
84	84	1	0Ah	Separador 0x0A
85	94	10	ASCII	Total de nota de débito (sin impuestos)
95	95	1	0Ah	Separador 0x0A
96	103	8	ASCII	Número de la Ultima Factura emitida
104	104	1	0Ah	Separador 0x0A
105	110	6	ASCII	Fecha de emisión de la Ultima Factura
111	111	1	0Ah	Separador 0x0A
112	115	4	ASCII	Hora de emisión de la Ultima Factura
116	116	1	0Ah	Separador 0x0A
117	124	8	ASCII	Número de Última Nota de Crédito
125	125	1	0Ah	Separador 0x0A
126	133	8	ASCII	Número de Última Nota de Débito
134	134	1	0Ah	Separador 0x0A
135	142	8	ASCII	Número del Último Documento No Fiscal
143	143	1	0Ah	Separador 0x0A



Librería de Clase **“TFHKADIR.DLL”** (Clase Protocolo Directo Tfhka)

Lectura del Reporte Z (U0Z)

Este comando permite leer desde el host (PC) el estado de la impresora fiscal, referente al reporte Z. Es posible ejecutar este comando en cualquier condición.

Desde	Hasta	Long	Clase	ITEM
2	5	4	ASCII	Número del Último Reporte Z emitido
6	6	1	0Ah	Separador 0x0A
7	12	6	ASCII	Fecha del último Reporte Z emitido
13	13	1	0Ah	Separador 0x0A
14	17	4	ASCII	Hora del último Reporte Z emitido
18	18	1	0Ah	Separador 0x0A
19	28	10	ASCII	Acumulado Exento
29	29	1	0Ah	Separador 0x0A
30	39	10	ASCII	Acumulado Impuesto Tasa 1
40	40	1	0Ah	Separador 0x0A
41	50	10	ASCII	Acumulado Impuesto Tasa 2
51	51	1	0Ah	Separador 0x0A
52	61	10	ASCII	Acumulado Impuesto Tasa 3
62	62	1	0Ah	Separador 0x0A
63	72	10	ASCII	Total de ventas (sin impuestos)
73	73	1	0Ah	Separador 0x0A
74	83	10	ASCII	Total de nota de crédito (sin impuestos)
84	84	1	0Ah	Separador 0x0A
85	94	10	ASCII	Total de nota de débito (sin impuestos)
95	95	1	0Ah	Separador 0x0A
96	103	8	ASCII	Número de la Última Factura emitida
104	104	1	0Ah	Separador 0x0A
105	110	6	ASCII	Fecha de emisión de la Última Factura
111	111	1	0Ah	Separador 0x0A
112	115	4	ASCII	Hora de emisión de la Última Factura
116	116	1	0Ah	Separador 0x0A
117	124	8	ASCII	Número de Última Nota de Crédito
125	125	1	0Ah	Separador 0x0A
126	133	8	ASCII	Número de Última Nota de Débito
134	134	1	0Ah	Separador 0x0A
135	142	8	ASCII	Número del Último Documento No Fiscal
143	143	1	0Ah	Separador 0x0A

Anexo 3:

Configuración del juego de caracteres en Linux.

Si se presenta el caso que el editor de código del fuente o la consola de la aplicación que va a implementar la clase no reconozcan algunos de los ASCII extendidos se debe realizar la configuración del juego de caracteres de Linux. Por defecto la codificación del juego de caracteres en LINUX viene en UTF-8, lo que genera inconveniente para el envío de ASCII extendidos al puerto serial, ya que los rechaza y no los procesa estando armada la trama de caracteres correctamente. Para ello se debe cambiar la configuración de los caracteres a ISO-8859-15 mediante la edición de estos archivos de la siguiente manera:

1. Cambiar el RC_LANG del archivo /etc/sysconfig/language.

vi /etc/sysconfig/language

RC_LANG="de_DE@euro"

2. Cambiar el CONSOLE_ENCODING del archivo /etc/sysconfig/console

vi /etc/sysconfig/console

CONSOLE_ENCODING="ISO-8859-15"

3. Agregar estas tres líneas de códigos en el archivo /etc/bash.bashrc

vi /etc/bash.bashrc.local

```
export LANG=de_DE@euro  
export LC_MESSAGES=POSIX  
export LC_CTYPE=de_DE@euro
```

4. Reinicie el sistema para que los cambios de la codificación de los caracteres tengan efecto.