

DLL MEGENCOM32

Especificação de Uso

Criação e produção:
Suporte Técnico em Julho/2014

Introdução.....	3
Instalação do driver DLPORTIO.SYS	3
Instalação da DLL	4
Arquivo de Log	4
Códigos de Retorno.....	4
Funções.....	6
MEGENCOM32_AbrirDispositivo()	6
MEGENCOM32_FecharDispositivo().....	8
MEGENCOM32_ConfigurarTimeoutsRXTX()	9
MEGENCOM32_EscrevernoDispositivo().....	10
MEGENCOM32_LeroDispositivo ()	12
MEGENCOM32_PreparaImpressao ().....	14
MEGENCOM32_AguardaImpressao ()	15
MEGENCOM32_AguardaImpressaoStr ().....	17
MEGENCOM32_StatusDoDispositivo ()	18
MEGENCOM32_AtivarLOG()	22
MEGENCOM32_DesativarLOG()	23

Introdução

Este documento descreve a forma de utilização da DLL MEGENCOM32, que é uma interface de comunicação com dispositivos DIEBOLD que possuem interfaces Serial, Paralela e/ou USB.

As funções descritas a seguir foram elaboradas no formato DLL (Dynamic Link Library) para ambientes Windows XP/Vista/7/8 32 bits permitindo o seu uso em aplicativos desenvolvidos em Visual Basic, Delphi, Visual C++ e outros.

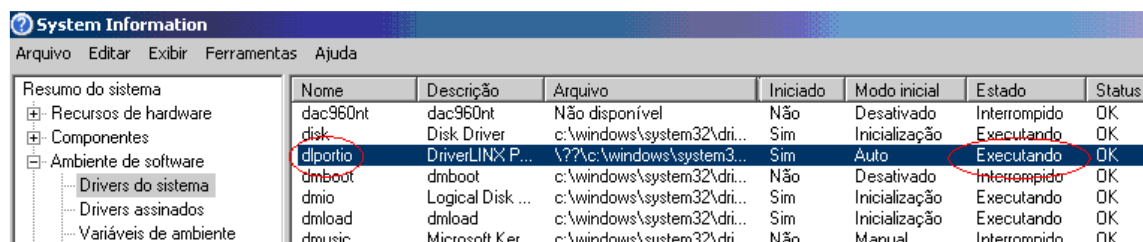
Instalação do driver DLPORTIO.SYS

O driver DLPORTIO.SYS é necessário somente para uso da DLL MEGENCOM32 com dispositivos DIEBOLD que possuem interface PARALELA nos ambientes Windows XP/Vista/7/8 32 bits.

Faça download do pacote de instalação do DLPORTIO na área de Download de nosso site. Use o INSTALL.EXE para fazer a instalação. É necessário estar logado como Administrador da máquina para realizar a instalação com sucesso.

Para saber se o DLPORTIO.SYS está instalado corretamente:

- Acesse: "Iniciar->Programas->Acessórios->Ferramentas do Sistema->Informações do Sistema";
- Localize "dlportio" na coluna "Nome";
- Verifique se a coluna "Estado" está "Executando":



Nome	Descrição	Arquivo	Iniciado	Modo inicial	Estado	Status
dac960nt	dac960nt	Não disponível	Não	Desativado	Interrompido	OK
disk	Disk Driver	c:\windows\system32\dri...	Sim	Inicialização	Executando	OK
dlportio	DriverLINX P...	\\??c:\windows\system3...	Sim	Auto	Executando	OK
dmboot	dmboot	c:\windows\system32\dri...	Não	Desativado	Interrompido	OK
dmio	Logical Disk ...	c:\windows\system32\dri...	Sim	Inicialização	Executando	OK
dmload	dmload	c:\windows\system32\dri...	Sim	Inicialização	Executando	OK
dmusic	Microsoft Ker...	c:\windows\system32\dri...	Não	Manual	Interrompido	OK

Caso tenha outras dúvidas, favor consultar o manual do DLPORTIO.

Instalação da DLL

Este pacote é composto por 4 arquivos do tipo DLL:

- MEGENCOM32: DLL genérica de comunicação Serial, Paralela e/ou USB;
- MECOM32: DLL de comunicação com interface Serial;
- MEPAR32: DLL de comunicação com interface Paralela;
- LIBPRINTIO: DLL de comunicação com interface USB;

Estas DLLs devem ser instaladas (copiadas) em uma pasta específica, conforme a plataforma utilizada:

Arquivos: MEGENCOM32.DLL, MECOM32.DLL, MEPAR32.DLL e LIBPRINTIO.DLL	
Plataforma Windows 32 bits	Diretório destino
Windows XP/Vista/7/8:	\Windows\System32
Para todas as versões:	Mesma pasta do seu aplicativo.

Lembre-se que todas estas DLLs devem ser instaladas (copiadas) na mesma pasta (todas juntas).

Arquivo de Log

O objetivo do arquivo de Log é fornecer uma ferramenta de auxílio ao desenvolvedor da aplicação que fará uso deste pacote. As funções, parâmetros e respostas trocadas entre o aplicativo e o dispositivo ficam registradas em um arquivo texto.

Este arquivo possui os comandos e funções executadas pela aplicação ao estabelecer comunicação com o dispositivo. Através deste arquivo é possível observar quais foram os comandos enviados e quais foram as respostas recebidas.

O nível de depuração garantirá o nível de detalhamento a ser gravado no arquivo de Log.

O arquivo de Log será gerado no diretório da aplicação ou diretório corrente da unidade em que a aplicação está sendo executada. Este arquivo é aberto e fechado a cada 10 segundos.

O nome do arquivo de Log é formado da seguinte maneira:

<AAAAMMDD>.MLG

Onde:

- AAAA – Representa o ano corrente.
- MM – O mês corrente e
- DD – O dia.

Por exemplo:

O arquivo <20030812.MLG> foi gerado em 12 de Agosto de 2003.

Códigos de Retorno

Constante	Valor	Descrição
MECAFCOD_RET_DLLNOTLOAD	-1	DLL não está carregada. A DLL ou função não foi encontrada.
MECAFCOD_RET_SUCESSO	0	Operação realizada com sucesso.
MECAFCOD_RET_FALHA	1	Falha na operação.
MECAFCOD_RET_ARGINVAL	2	Argumentos inválidos.
MECAFCOD_RET_ERRALOCMEM	3	Falha na alocação de memória.
MECAFCOD_RET_ERRLOCKMEM	4	Não conseguiu restringir acesso a uma região de memória.
MECAFCOD_RET_TIMEOUT	5	Timeout durante realização da operação.
MECAFCOD_RET_ERRCOMUNICACAO	6	Erro durante o processo de comunicação
MECAFCOD_RET_CMDEXEC	7	Existe uma função, ou comando, em execução.
MECAFCOD_RET_ARQINIINVAL	8	Arquivo de configuração (.INI) inválido.
MECAFCOD_RET_DISPNOTOPEN	9	Dispositivo não foi aberto.
MECAFCOD_RET_ERROPENSER	10	Erro ao abrir dispositivo serial.
MECAFCOD_RET_ERRCONF	11	String de configuração do dispositivo inválido.
MECAFCOD_RET_PROGSER	12	Erro ao configurar a porta serial.
MECAFCOD_RET_DEFTAMBUF	13	Erro ao definir tamanho dos buffers de recepção e transmissão.
MECAFCOD_RET_PROGTIMEOUTS	14	Erro ao configurar os valores de timeout para operações de envio e recepção.
MECAFCOD_RET_EVENTREAD	15	Erro na criação do evento de leitura.
MECAFCOD_RET_EVENTWRITE	16	Erro na criação do evento de escrita.
	17 a 19	Reservado para uso futuro.
MECAFCOD_RET_ARGPORTA	20	Valor da porta inválido.
MECAFCOD_RET_ARGVELOC	21	Valor da velocidade de transmissão inválido.
MECAFCOD_RET_ARGPARID	22	Valor da paridade inválido.
MECAFCOD_RET_ARGNUMBITS	23	Valor do número de bits inválido.
MECAFCOD_RET_ARGSTOPBITS	24	Valor do número de stop bits inválido.
	25 a 79	Reservado para uso futuro.
MECAFCOD_RET_DISPOPEN	80	Dispositivo já está aberto.
MECAFCOD_RET_NUMDISPINVAL	81	Número máximo de dispositivos foi excedido.
MECAFCOD_RET_CMDNOTFOUND	82	Comando inexistente.
MECAFCOD_RET_DRIVERBUSY	83	Existem JOBs pendentes.
MECAFCOD_RET_EOP	84	Ocorreu fim de papel durante a impressão.

Funções

Função

MEGENCOM32_AbrirDispositivo()

Descrição: A função MEGENCOM32_AbrirDispositivo() é utilizada abrir e configurar a porta de comunicação onde o dispositivo DIEBOLD está conectado. Basta chamá-la uma única vez ao iniciar o aplicativo.

Declarações (Prototype):

C: int MEGENCOM32_AbrirDispositivo(char *Porta, DWORD Velocidade, char Paridade, int NumBits, int NumStopbits, int ControleDeFluxo);

Visual Basic: Declare Function MEGENCOM32_AbrirDispositivo Lib "MEGENCOM32.dll" (ByVal Porta As String, ByVal Velocidade As Long, ByVal Paridade As Byte, ByVal NumBits As Long, ByVal NumStopbits As Long, ByVal CtrlFluxo As Long) As Long

Delphi: function MEGENCOM32_AbrirDispositivo(Porta: PChar; Velocidade: LongInt; Paridade: Byte; NumBits, NumStopBits, ControleDeFluxo : LongInt): longInt; StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação.

Opções:

- Serial: "COM1" a "COM99";
- Paralela: "LPT1", "LPT2";
- USB: "USB000", "USB001", "USB002", ... , "USB009"

Note que "USB000" deve ser usado para detecção automática de porta USB utilizada por dispositivo DIEBOLD.

Velocidade: Velocidade (BaudRate) da comunicação serial(*).

Opções:

110, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200, 128000, 256000, 230400, 460800 (bps).

Paridade: Paridade da comunicação serial (*).

Opções:

'P' – Paridade par
'I' – Paridade ímpar
'S' – Sem paridade

NumBits: Número de Bits da comunicação serial (*).

Opções:

8 - bits
7 - bits

NumStopbits: Número de Stop Bits da comunicação serial (*).

Opções:

1 - Stop Bit

2 - Stop Bits

ControleDeFluxo: Controle de Fluxo (protocolo) da comunicação serial (*).

Opções:

0 - Nenhum

1 - CTS/RTS (Hardware)

2 - XOn/XOff (Software)

(*) Para interfaces Paralela e USB deve-se usar o valor 0 (zero) para esses parâmetros.

Exemplo em C:

```
int iRet;
iRet = MEGENCOM32_AbrirDispositivo( "COM1", 9600, 'S', 8, 1, 1 );
if( iRet != MECAFCOD_RET_SUCESSO )
{
    // Tratamento em caso de Erro
    return( iRet );
}
// Tratamento em caso de Sucesso
return( iRet );
```

Exemplo em Visual Basic:

```
Function Teste()
Dim iRet as Integer
iRet = MEGENCOM32_AbrirDispositivo( "LPT1" ,0 ,0 ,0 ,0 ,0 )
If iRet <> MECAFCOD_RET_SUCESSO Then
    ` Tratamento em caso de Erro
    Exit Function
End If
` Tratamento em caso de Sucesso
Exit Funtion
```

Exemplo em Delphi:

```
Function AbrirPorta(): Boolean;
var
    iRet: Integer;
begin
    Result := false;
    iRet := MEGENCOM32_AbrirDispositivo( "USB000" ,0 ,0 ,0 ,0 ,0 );
    if iRet <> MECAFCOD_RET_SUCESSO then
        begin
            // Tratamento em caso de Erro
            exit;
        end;
    // Tratamento em caso de Sucesso
    Result := true;
End;
```

Função	<i>MEGENCOM32_FecharDispositivo()</i>
---------------	--

Descrição: A função MEGENCOM32_FecharDispositivo() encerra o uso da porta de comunicação por essa API e libera o uso da porta para outros aplicativos. Basta ser chamada uma única vez ao encerrar o aplicativo.

Declarações (Prototype):

C: int MEGENCOM32_FecharDispositivo(char *Porta);

Visual Basic: Declare Function MEGENCOM32_FecharDispositivo Lib "MEGENCOM32.dll" (ByVal Porta As String) As Long

Delphi: function MEGENCOM32_FecharDispositivo(Porta: PChar): Integer;
StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação que deseja encerrar o uso.

Opções:

- Veja **Porta** da função MEGENCOM32_AbrirDispositivo()

Exemplo em C:

```
int iRet;  
iRet = MEGENCOM32_FecharDispositivo( "COM1" );  
if( iRet != MECAFCOD_RET_SUCESSO )  
{  
    // Tratamento em caso de Erro  
    return( iRet );  
}  
// Tratamento em caso de Sucesso  
return( iRet );
```

Exemplo em Visual Basic:

```
Dim iRet as Integer  
iRet = MEGENCOM32_FecharDispositivo( "LPT1" )  
If iRet <> MECAFCOD_RET_SUCESSO Then  
    ' Tratamento em caso de Erro  
    Exit Function  
End If  
' Tratamento em caso de Sucesso  
Exit Funtion
```

Exemplo em Delphi:


```
Function FecharPorta(): Boolean;  
var  
    iRet: Integer;  
begin  
    Result := false;  
    iRet := MEGENCOM32_FecharDispositivo( "USB000" ,0 ,0 ,0 ,0 ,0 );  
    if iRet <> MECAFCOD_RET_SUCESSO then  
        begin  
            // Tratamento em caso de Erro  
            exit;  
        end;  
    // Tratamento em caso de Sucesso  
    Result := true;  
end;
```

Função	<i>MEGENCOM32_ConfigurarTimeoutsRXTX()</i>
---------------	---

Descrição: A função MEGENCOM32_ConfigurarTimeoutsRXTX() pode ser usada para configurar o tempo (em milisegundos) de intervalo máximo entre bytes nas operações de escrita e leitura.

Declarações (Prototypes):

C: int MEGENCOM32_ConfigurarTimeoutsRXTX(char *Porta, DWORD TimeOutRX, DWORD TimeOutTX);

Visual Basic: Declare Function MEGENCOM32_ConfigurarTimeoutsRXTX Lib "MEGENCOM32.dll" (ByVal Porta As String, ByVal TimeOutRX As Long, ByVal TimeOutTX As Long) As Long

Delphi: function MEGENCOM32_ConfigurarTimeoutsRXTX(Porta: PChar; TimeOutRX, iTimeOutTX: LongInt): Integer; StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação.

Opções:

- Veja **Porta** da função MEGENCOM32_AbrirDispositivo().

TimeOutRX: Tempo entre bytes para operação de leitura (em milisegundos).

TimeOutTX: Tempo entre bytes para operação de escrita (em milisegundos).

Exemplo em C:

```
int iRet;  
iRet = MEGENCOM32_ConfigurarTimeoutsRXTX ( "COM1", 2000, 5000 );  
if( iRet != MECAFCOD_RET_SUCESSO )  
{  
    // Tratamento em caso de Erro  
    return( iRet );  
}  
// Tratamento em caso de Sucesso  
return( iRet );
```

Exemplo em Visual Basic:

```
Function Teste()  
    Dim iRet as Integer  
    iRet = MEGENCOM32_ConfigurarTimeoutsRXTX( "LPT1", 2000, 5000 )  
    If iRet <> MECAFCOD_RET_SUCESSO Then  
        ' Tratamento em caso de Erro  
        Exit Function  
    End If  
    ' Tratamento em caso de Sucesso  
End Function
```

Exemplo em Delphi:

```
Function Teste(): boolean;  
var  
    iRet: Integer;  
begin  
    Result := True;  
    iRet:= MEGENCOM32_ConfigurarTimeoutsRXTX('USB000',2000, 5000 );  
    if (iRet <> MECAFCOD_RET_SUCESSO) Then  
        Begin  
            // Tratamento em caso de Erro  
            Result := False;  
            exit;  
        end;  
        // Tratamento em caso de Sucesso  
    End;  
End;
```

Função	<i>MEGENCOM32_EscrevernoDispositivo()</i>
---------------	--

Descrição: A função MEGENCOM32_EscrevernoDispositivo() realiza a impressão de um buffer contendo um texto com ou sem formatação. O buffer também poderá conter qualquer seqüência de ESC disponível na tabela de comandos do dispositivo DIEBOLD, como por exemplo, comandos para controle de atributos de impressão, Modo Gráfico, solicitação de status, etc.

Declarações (Prototypes):

```
C: int MEGENCOM32_EscrevernoDispositivo( char *Porta, BYTE *Buffer,  
    DWORD NumBytes, DWORD *BytesEscritos );
```

Visual Basic: Declare Function MEGENCOM32_EscrevernoDispositivo Lib "MEGENCOM32.dll" (ByVal Porta As String, ByVal Buffer As String, ByVal NumBytes As Long, ByRef BytesEscritos As Long) As Long

Delphi: Function MEGENCOM32_EscrevernoDispositivo(Porta: PChar; Buffer: PChar; NumBytes: LongInt; NumBytesEscritos: PLong): Integer; StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação.

Opções:

- Veja **Porta** da função MEGENCOM32_AbrirDispositivo().

Buffer: Ponteiro para o Buffer a ser impresso ou enviado ao dispositivo.

NumBytes: Número de bytes a serem impressos ou enviados ao dispositivo.

BytesEscritos: Ponteiro para receber o número de bytes efetivamente enviados ao dispositivo.

OBS: Em caso de erro, somente a interface Paralela irá preencher *BytesEscritos*.

Exemplo em C:

```
int iRet;
DWORD BytesEscritos;
iRet = MEGENCOM32_EscrevernoDispositivo( "COM1", "Texto a ser impresso.",
21, &BytesEscritos );
if( iRet != MECAFCOD_RET_SUCESSO )
{
    // Tratamento em caso de Erro
    return( iRet );
}
// Tratamento em caso de Sucesso
return( iRet );
```

Exemplo em Visual Basic:

```
Function Teste()  
  Dim iRet as Integer  
  Dim BytesEscritos As Long  
  Dim bBuffer As String  
  Dim cmdAtivaNegrito As String  
  Dim cmdDesativaNegrito As String  
  
  cmdAtivaNegrito = Chr(27) & "E"  
  cmdDesativaNegrito = Chr(27) & "F"  
  bBuffer = cmdAtivaNegrito & "Negrito (ênfâtizado)" & cmdDesativaNegrito &  
  Chr(10)  
  iRet = MEGENCOM32_EscrevernoDispositivo( "LPT1", bBuffer, 24,  
  BytesEscritos )  
  If iRet <> MECAFCOD_RET_SUCESSO Then  
    ` Tratamento em caso de Erro  
    Exit Function  
  End If  
  ` Tratamento em caso de Sucesso  
End Function
```

Exemplo em Delphi:

```
Function Teste(): boolean;  
var  
  iRet: Integer;  
  NumBytesEscritos: LongInt;  
begin  
  Result := True;  
  
  iRet:= MEGENCOM32_EscrevernoDispositivo('USB000', 'Texto a ser impresso.',  
  21,@NumBytesEscritos);  
  if (iRet <> MECAFCOD_RET_SUCESSO) Then  
    Begin  
      // Tratamento em caso de Erro  
      Result := False;  
      Exit;  
    End;  
    // Tratamento em caso de Sucesso  
  End;  
End;
```

Função***MEGENCOM32_LeroDispositivo ()***

Descrição: Lê dados do dispositivo DIEBOLD. Poderá ser lido um único byte de resposta de solicitação de Status ou uma seqüência de bytes.

Declarações (Prototypes):

C: Int MEGENCOM32_LeroDispositivo(char *Porta, BYTE *Buffer, DWORD
 NumBytes, DWORD *BytesLidos);

Visual Basic: Declare Function MEGENCOM32_LeroDispositivo Lib "MEGENCOM32.dll"
(ByVal Porta As String, ByVal Buffer As String, ByVal NumBytes As
Long, ByRef BytesLidos As Long) As Long

Delphi: function MEGENCOM32_LeroDispositivo(Porta: PChar; Buffer: PChar;
NumBytes: LongInt; NumBytesLidos: PInteger): Integer; StdCall;
External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação.

Opções:

- Veja **Porta** da função MEGENCOM32_AbrirDispositivo().

Buffer: Ponteiro para receber os dados lidos do dispositivo.

NumBytes: Número de bytes (tamanho) disponíveis para o Buffer de leitura.

BytesLidos: Ponteiro para receber o número de bytes efetivamente lidos do dispositivo.

Exemplo em C:

```
int iRet;  
char bBuffer[5];  
DWORD BytesLidos;  
iRet = MEGENCOM32_LeroDispositivo( "COM1", &bBuffer[0], 5, &BytesLidos);  
if( iRet != MECAFCOD_RET_SUCESSO )  
{  
    // Tratamento em caso de Erro  
    return( iRet );  
}  
// Tratamento em caso de Sucesso  
return( iRet );
```

Exemplo em Visual Basic:

```
Private Function Teste()  
    Dim iRet as Integer  
    Dim BytesLidos As Long  
    Dim bBuffer As String  
    iRet = MEGENCOM32_LeroDispositivo ( "LPT1", bBuffer, 1, BytesLidos )  
    If iRet <> MECAFCOD_RET_SUCESSO Then  
        ` Tratamento em caso de Erro  
        Exit Function  
    End If  
    ` Tratamento em caso de Sucesso  
End Function
```

Exemplo em Delphi:

```
Function Teste(): boolean;  
var  
  iRet: Integer;  
  NumBytesLidos: LongInt;  
  bBuffer: array [0..1] of Char;  
begin  
  Result := True;  
  iRet:= MEGENCOM32_LeroDispositivo ('USB000', bBuffer, 1,  
@NumBytesLidos );  
  if (iRet <> MECAFCOD_RET_SUCESSO) Then  
    Begin  
      // Tratamento em caso de Erro  
      Result := False;  
      exit;  
    end;  
End;
```

Função	<i>MEGENCOM32_PreparaImpressao ()</i>
---------------	--

Descrição: A função MEGENCOM32_PreparaImpressao() prepara controles internos da DLL MEGENCOM32 para uma nova impressão. Caso queira usar esse método, a sequência de execução deverá ser:

- 1º - MEGENCOM32_PreparaImpressao()
- 2º - MEGENCOM32_EscrevernoDispositivo()
- 3º - MEGENCOM32_AguardaImpressao()

Ou

- 3º - MEGENCOM32_AguardaImpressaoStr()

Declarações (Prototypes):

C: int MEGENCOM32_PreparaImpressao(char *Porta);

Visual Basic: Declare Function MEGENCOM32_PreparaImpressao Lib
"MEGENCOM32.dll" (ByVal Porta As String) As Long

Delphi: function MEGENCOM32_PreparaImpressao(Porta: PChar): Integer;
StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação.

Opções:

- Veja **Porta** da função MEGENCOM32_AbrirDispositivo().

Exemplo em C:

```
int iRet;  
iRet = MEGENCOM32_PreparaImpressao ( "COM1" );  
if( iRet != MECAFCOD_RET_SUCESSO )  
{  
    // Tratamento em caso de Erro  
    return( iRet );  
}  
// Tratamento em caso de Sucesso  
return( iRet );
```

Exemplo em Visual Basic:

```
Function Teste()  
Dim iRet as Integer  
iRet = MEGENCOM32_PreparaImpressao ( "LPT1" )  
If iRet <> MECAFCOD_RET_SUCESSO Then  
    ` Tratamento em caso de Erro  
    Exit Function  
End If  
` Tratamento em caso de Sucesso  
End Function
```

Exemplo em Delphi:

```
Function Prepara_Impressao (): boolean;  
var  
    iRet: Integer;  
begin  
    Result := True;  
    iRet:= MEGENCOM32_PreparaImpressao ('USB000');  
    if (iRet <> MECAFCOD_RET_SUCESSO) Then  
        Begin  
            // Tratamento em caso de Erro  
            Result := False;  
            exit;  
        end;  
    // Tratamento em caso de Sucesso  
End;
```

Função	<i>MEGENCOM32_AguardaImpressao ()</i>
---------------	--

Descrição: A função MEGENCOM32_AguardaImpressao() pode ser usada para aguardar e confirmar se uma impressão foi realizada com sucesso. Caso queira usar esse método, a sequência de execução deverá ser:

- 1º - MEGENCOM32_PreparaImpressao()
- 2º - MEGENCOM32_EscrevernoDispositivo()
- 3º - MEGENCOM32_AguardaImpressao()

Ou

- 3º - MEGENCOM32_AguardaImpressaoStr()

Declarações (Prototypes):

C: int MEGENCOM32_AguardaImpressao(char *Porta , DWORD
Timeout);

Visual Basic: Declare Function MEGENCOM32_AguardaImpressao Lib
"MEGENCOM32.dll" (ByVal Porta As String, ByVal Timeout As Long) As
Long

Delphi: function MEGENCOM32_AguardaImpressao(Porta: PChar; Timeout:
LongInt): Integer; StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação.

Opções:

- Veja **Porta** da função MEGENCOM32_AbrirDispositivo().

Timeout: Tempo de espera em milissegundos. Deve ser calculado em
função do número de bytes enviados através da função
MEGENCOM32_EscrevernoDispositivo().

Exemplo em C:

```
int iRet;  
iRet = MEGENCOM32_AguardaImpressao ( "COM1", 4000 );  
if( iRet != MECAFCOD_RET_SUCESSO )  
{  
    // Tratamento em caso de Erro  
    return( iRet );  
}  
// Tratamento em caso de Sucesso  
return( iRet );
```

Exemplo em Visual Basic:

```
Function Teste()  
Dim iRet as Integer  
iRet = MEGENCOM32_AguardaImpressao ( "LPT1", 4000 )  
If iRet <> MECAFCOD_RET_SUCESSO Then  
    ` Tratamento em caso de Erro  
    Exit Function  
End If  
` Tratamento em caso de Sucesso  
End Function
```

Exemplo em Delphi:


```
Function Aguarda_Impressao (): boolean;  
var  
    iRet: Integer;  
begin  
    Result := True;  
    iRet:= MEGENCOM32_AguardaImpressao ('USB000', 4000);  
    if (iRet <> MECAFCOD_RET_SUCESSO) Then  
        Begin  
            // Tratamento em caso de Erro  
            Result := False;  
            Exit;  
        End;  
    // Tratamento em caso de Sucesso  
End;
```

Função	<i>MEGENCOM32_AguardaImpressaoStr ()</i>
---------------	---

Descrição: A função MEGENCOM32_AguardaImpressaoStr() é parecida com função MEGENCOM32_AguardaImpressao(). A única diferença está no tipo do parâmetro *Timeout*.

Declarações (Prototypes):

C: int MEGENCOM32_AguardaImpressaoStr(char *Porta ,char *Timeout);

Visual Basic: Declare Function MEGENCOM32_AguardaImpressao Lib "MEGENCOM32.dll" (ByVal Porta As String, ByVal Timeout As String) As Long

Delphi: function MEGENCOM32_AguardaImpressao(Porta: PChar; Timeout: PChar): Integer; StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação.

Opções:

- Veja **Porta** da função MEGENCOM32_AbrirDispositivo().

Timeout: Tempo de espera em milissegundos. Deve ser calculado em função do número de bytes enviados através da função MEGENCOM32_EscrevernoDispositivo().

Exemplo em C:

```
int iRet;  
iRet = MEGENCOM32_AguardaImpressao ( "COM1", "4000" );  
if( iRet != MECAFCOD_RET_SUCESSO )  
{  
    // Tratamento em caso de Erro  
    return( iRet );  
}  
// Tratamento em caso de Sucesso  
return( iRet );
```

Exemplo em Visual Basic:

```
Function Teste()  
Dim iRet as Integer  
iRet = MEGENCOM32_AguardaImpressao ( 'LPT1', '4000' )  
If iRet <> MECAFCOD_RET_SUCESSO Then  
    ' Tratamento em caso de Erro  
    Exit Function  
End If  
' Tratamento em caso de Sucesso  
End Function
```

Exemplo em Delphi:

```
Function Aguarda_Impressao (): boolean;  
var  
    iRet: Integer;  
begin  
    Result := True;  
    iRet:= MEGENCOM32_AguardaImpressao ('USB000', '4000');  
    if (iRet <> MECAFCOD_RET_SUCESSO) Then  
        Begin  
            // Tratamento em caso de Erro  
            Result := False;  
            Exit;  
        End;  
    // Tratamento em caso de Sucesso  
End;
```

Função	MEGENCOM32_StatusDoDispositivo ()
---------------	--

Descrição: A função MEGENCOM32_StatusDoDispositivo() pode ser usada para ler o status da porta de comunicação do dispositivo. Lembre-se que há outras maneiras de se obter o status do dispositivo DIEBOLD, por exemplo, através dos status imediato, bufferizado, cíclico e único em caso de erro. Para saber mais detalhes, consulte o manual do firmware e os programas exemplos.

Declarações (Prototypes):

C: int MEGENCOM32_StatusDoDispositivo(char *Porta, BYTE *Status);

Visual Basic: Declare Function MEGENCOM32_StatusDoDispositivo Lib
"MEGENCOM32.dll" (ByVal Porta As String, ByRef Status As Byte) As
Long

Delphi: function MEGENCOM32_StatusDoDispositivo(Porta: PChar; Status :
PChar): Integer; StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Porta: Nome da Porta de Comunicação.

Opções:

- Veja **Porta** da função MEGENCOM32_AbrirDispositivo().

Status: Ponteiro para o buffer que receberá o valor do status. Esse byte
deverá ser validado somente se o retorno dessa função for
MECAFCOD_RET_SUCESSO:

Interface Serial:

- Bit 7: Em 1 indica que a impressora está Busy
- Bit 6: Reservado
- Bit 5: Reservado
- Bit 4: Em 1 indica que a impressora está Off Line
- Bit 3: Reservado
- Bit 2: Reservado
- Bit 1: Reservado
- Bit 0: Reservado

Interface Paralela:

- Bit 7: Em 1 indica que a impressora está Busy
- Bit 6: Reservado
- Bit 5: Em 1 indica que a impressora está Sem Papel
- Bit 4: Em 1 indica que a impressora está On Line
- Bit 3: Em 1 indica que a impressora está Sem Erro
- Bit 2: Reservado
- Bit 1: Reservado
- Bit 0: Reservado

Interface USB:

- Esta função não está implementada para Interface USB,
portanto o retorno será sempre
MECAFCOD_RET_CMDNOTFOUND
- Não validar o retorno do parâmetro Status

Exemplo em C:

```
#define BIT0      0x01
#define BIT1      0x02
#define BIT2      0x04
#define BIT3      0x08
#define BIT4      0x10
#define BIT5      0x20
#define BIT6      0x40
#define BIT7      0x80
int iRet;
BYTE bStatus;
iRet = MEGENCOM32_StatusDoDispositivo ( "COM1", &bStatus)
if( iRet == MECAFCOD_RET_SUCESSO )
{
    if((bStatus & BIT7)!=0)
        // Impressora Busy
    else
        // Impressora Not Busy

    if((bStatus & BIT4) ==0)
        // Impressora On Line
    else
        // Impressora Off Line
}
return( iRet )
```

Exemplo em Visual Basic:

```
Const BIT0 = 1           ' Mascara p/ Verificar Bit 0
Const BIT1 = 2           ' Mascara p/ Verificar Bit 1
Const BIT2 = 4           ' Mascara p/ Verificar Bit 2
Const BIT3 = 8           ' Mascara p/ Verificar Bit 3
Const BIT4 = 16          ' Mascara p/ Verificar Bit 4
Const BIT5 = 32          ' Mascara p/ Verificar Bit 5
Const BIT6 = 64          ' Mascara p/ Verificar Bit 6
Const BIT7 = 128        ' Mascara p/ Verificar Bit 7
Function Teste()
Dim iRet As Integer
Dim bStatus As Byte
iRet = MEGENCOM32_StatusDoDispositivo(PortaComunicacao, bStatus)
If iRet = 0 Then
    If (bStatus And BIT3) = 0 Then
        ' Impressora com Erro
    Else
        ' Impressora Sem Erro
    End If

    If (bStatus And BIT4) = 0 Then
        ' Impressora Off Line
    Else
        ' Impressora On Line
    End If

    If (bStatus And BIT5) = 0 Then
        ' Impressora Com Papel
    Else
        ' Impressora Sem Papel
    End If

    If (bStatus And BIT7) = 0 Then
        ' Impressora Not Busy
    Else
        ' Impressora Busy
    End If
End If
End Function
```

Exemplo em Delphi:

```
Function Teste(): boolean;  
var  
    iRet: Integer;  
    bStatus: Integer;  
begin  
    Result := True;  
    iRet:= StatusDoDispositivo ( "USB000", @bStatus)  
    if (iRet <> MECAFCOD_RET_SUCESSO) Then  
        Begin  
            // Tratamento em caso de Erro  
            Result := False;  
            exit;  
        end;  
    // Tratamento em caso de Sucesso  
    // Lembre-se que por enquanto esta função não está implementada na  
    // interface USB  
End;
```

Função	MEGENCOM32_AtivarLOG()
---------------	-------------------------------

Descrição: A função MEGENCOM32_AtivarLOG é usada para ativar a gravação de Log.

Declarações (Prototypes):

C: Int MEGENCOM32_AtivarLOG(int Camada ,int Nivel);

Visual Basic: Declare Function MEGENCOM32_AtivarLOG Lib "MEGENCOM32.dll"
(ByVal Camada As Integer, ByVal Nivel As Integer) As Long

Delphi: function MEGENCOM32_AtivarLOG(Camada, Nivel: Integer): Integer;
StdCall; External 'MEGENCOM32.DLL';

Parâmetros:

Camada: Escolha a camada do Log que deseja ativar:

- 0: Camada Genérica
- 1: Camada Serial
- 2: Camada Paralela
- 3: Todas as camadas
- 4: Camada USB

Nivel: Escolha o nível de gravação que deseja ativar:

- 1: Nível Básico
- 2: Nível Médio
- 3: Nível Avançado

Exemplo em C:

```
int iRet;  
iRet = MEGENCOM32_AtivarLOG ( 3, 2 );  
if( iRet != MECAFCOD_RET_SUCESSO )  
{  
    // Tratamento em caso de Erro  
    return( iRet );  
}  
// Tratamento em caso de Sucesso  
return( iRet );
```

Exemplo em Visual Basic:

```
Function Teste()  
Dim iRet as Integer  
iRet = MEGENCOM32_AtivarLOG ( 3, 2 )  
If iRet <> MECAFCOD_RET_SUCESSO Then  
    ` Tratamento em caso de Erro  
    Exit Function  
End If  
` Tratamento em caso de Sucesso  
End Function
```

Exemplo em Delphi:

```
Function Ativar_Log (): boolean;  
var  
    iRet: Integer;  
begin  
    Result := True;  
    iRet:= MEGENCOM32_AtivarLOG ( 3, 2 );  
    if (iRet <> MECAFCOD_RET_SUCESSO) Then  
        Begin  
            // Tratamento em caso de Erro  
            Result := False;  
            Exit;  
        End;  
    // Tratamento em caso de Sucesso  
End;
```

Função	<i>MEGENCOM32_DesativarLOG()</i>
---------------	---

Descrição: A função MEGENCOM32_DesativarLOG() é usada para desativar a gravação de Log.

Declarações (Prototypes):

C: int MEGENCOM32_DesativarLOG(int Camada);

Visual Basic: Declare Function MEGENCOM32_DesativarLOG Lib "MEGENCOM32.dll"
(ByVal Camada As Integer) As Long

Delphi: MEGENCOM32_DesativarLOG(Camada: Integer): Integer; StdCall;
External 'MEGENCOM32.dll';

Parâmetros:

Camada: Escolha a camada do Log que deseja desativar:

- 0: Camada Genérica
- 1: Camada Serial
- 2: Camada Paralela
- 3: Todas as camadas
- 4: Camada USB

Exemplo em C:

```
int iRet;  
iRet = MEGENCOM32_DesativarLOG ( 3 );  
if( iRet != MECAFCOD_RET_SUCESSO )  
{  
    // Tratamento em caso de Erro  
    return( iRet );  
}  
// Tratamento em caso de Sucesso  
return( iRet );
```

Exemplo em Visual Basic:

```
Function Teste()  
Dim iRet as Integer  
iRet = MEGENCOM32_DesativarLOG ( 3 )  
If iRet <> MECAFCOD_RET_SUCESSO Then  
    ` Tratamento em caso de Erro  
    Exit Function  
End If  
` Tratamento em caso de Sucesso  
End Function
```

Exemplo em Delphi:

```
Function Desativar_Log (): boolean;  
var  
    iRet: Integer;  
begin  
    Result := True;  
    iRet:= MEGENCOM32_DesativarLOG ( 3 );  
    if (iRet <> MECAFCOD_RET_SUCESSO) Then  
        Begin  
            // Tratamento em caso de Erro  
            Result := False;  
            Exit;  
        End;  
    // Tratamento em caso de Sucesso  
End;
```