

Manual Funções DLL



Manual

DT433

Introdução:	3
--------------------	----------

Funções Padrão	4
-----------------------	----------

Function InicializaSerial (nporta:byte): Boolean;	5
Function InicializaSocket (ip:pchar): Boolean;	5
Function FechaSerial: DWORD;	5
Function FechaSocket: boolean;	6
Function LeAbastecimento:abast;	6
Function LeAbastecimentoPAF1(autoIncrement:boolean=true):AbastPAF1;	6
Procedure Incrementa;	7
Function LeVisualizacao():OnLine;	7
Function LeStatus():multistatus;	7
Function AutoLibera (bico:string): Error;	8
Function AutorizaAbast (bico:string): Error;	8
Function BloqueiaBico (bico:string): Error;	8
Function ParaBomba (bico:string): Error;	9
Function AlteraPreco (bico:string; preco:double; decimais:byte): error;	9
Function Preset (bico:string; valor:double): Error;	9
Function ConsultaEncerrante (modo:char; bico:string): Encerrante;	10
Function getclock:string;	10
Function SetClock(par:string):boolean;	11
Function GetVisualizaoid:shortstring;	11
Function FidIdent:IFid;	11
Procedure FidIncrementa;	12
Function saveTagFid (controle1,controle2:char; tag,t1in,t1out,t2in,t2out:string): integer;..	12
Function deleteTagFid (posicao:integer; tag:string): integer;	13
procedure clearTagFid;	13
Function FidLeRegistro (nro:integer): ShortString;	14
Function SetBlackList (modo:char; TAG:string): boolean;	14
Function PresetId (bico,tag:string; cliente,libera:bool; valpreset,tempo:integer): Boolean;.	15

Funções compatíveis com VB6	15
------------------------------------	-----------

Function VB_OpenSerial(np: integer):integer;	15
Function VB_OpenSocket(ip: string):integer;	16
Function VB_CloseSerial:integer;	16
Function VB_CloseSocket:integer;	16
Function VB_ReadSale(var st:string):integer;	17
Function VB_ReadSalePAF(var st:string):integer;	17
function VB_NextSale:integer;	17
function VB_ReadState(var st:string):integer;	18
Function VB_AutPump(bico:string):integer;	18
Function VB_BlockPump(bico:string):integer;	18
Function VB_FreePump(bico:string):integer;	19
Function VB_StopPump(bico: String):integer;	19
function VB_PresetPump(bico,cash:string):integer;	19
function VB_SetPrice(bico,price:pchar):integer;	20

function VB_ReadTotalsCash(bico:pchar):integer;	20
function VB_ReadTotalsVolume(bico:string):integer;	20
function VB_SendReceiveText(var st:string;timeout:integer):integer;	21
Funções Compatíveis com C/C++/C#/VB.NET	21
function C_OpenSerial(np:integer):integer;	22
function C_OpenSocket(ip:pchar):integer;	22
function C_CloseSerial:integer;	23
Function C_CloseSocket:integer;	23
Function C_GetSalePAF():pchar;	23
Procedure C_NextSale;	24
function C_readState():pchar;	24
function C_AutoPump(bico:pchar):integer;	24
function C_BlockPump(bico:pchar):integer;	25
function C_FreePump(bico:pchar):integer;	25
function C_PresetPump(bico,cash:pchar):integer;	26
function C_ReadTotalsVolume(bico:pchar):integer;	27
Function C_Visualize:pchar;	27
Function C_GetClock:pchar;	27
Function C_SetClock(par:pchar):boolean;	28
Function C_SendReceiveText(comando:shortstring):shortstring;	28
Estruturas de dados.....	29

Introdução:

Caro desenvolvedor. Essa biblioteca foi desenvolvida para facilitar a implementação dos equipamentos Companytec em seu software. Disponibilizando todas as funções necessárias para a integração, nossa DLL já está presente em aproximadamente 40% dos sistemas compatíveis com nossa solução.

A fim de atender as linguagens mais utilizadas na atualidade, nossa DLL possui funções que se adaptam à sua linguagem, para que isso seja possível, existem várias funções que executam o mesmo trabalho, diferenciando-se apenas na sua maneira de passagem de parâmetros e retornos. Entre as linguagens que já utilizam a DLL podemos citar:

- Delphi
- Visual Basic 6
- Visual Basic .NET

- C#
- C/C++
- Fox Pro
- COBOL for Windows.

Funções Padrão

O conjunto de funções apresentadas neste capítulo representam as primeiras e principais funções criadas nesta DLL, caso o desenvolvedor tenha problemas em tratar o parâmetros passados pelas funções e problemas de tipagem de dados, como a dificuldade em criar as estruturas de dados definidas na DLL, o desenvolvedor poderá optar por outras soluções apresentadas nos capítulos abaixo.

As funções definidas na DLL não são específicas por linguagem, logo o desenvolvedor pode utilizar algumas funções deste capítulo, e quaisquer outra função que ele tiver dificuldade, poderá optar por escolher a função de outros capítulos.

As funções definidas nos próximos capítulos, por exemplo VB6, podem ser utilizados por outras linguagens de programação sem qualquer problema, a criação destes conjuntos de funções específicas se deu em função de restrições impostas pela linguagem alvo.

Function InicializaSerial (nporta:byte): Boolean;

Função que abre a porta de comunicações para envio e recebimento de comandos.

Essa função só necessita ser chamada uma vez, no início da aplicação que irá comunicar com nosso equipamento.

Parâmetros de entrada:

- nporta: Número da porta serial

BYTE

Retorno da função:

- “True” se conectou ou “False” se não foi possível se conectar;

BOOLEAN

Function InicializaSocket (ip:pchar): Boolean;

Função que abre a porta de comunicações para envio e recebimento de comandos.

Essa função só necessita ser chamada uma vez, no início da aplicação que irá comunicar com nosso equipamento.

Parâmetros de entrada:

- ip: Endereço na rede em que será feito a conexão

PCHAR

Retorno da função:

- “True” se conectou ou “False” se não foi possível se conectar;

BOOLEAN

Function FechaSerial: DWORD;

Função chamada para finalizar uma conexão do tipo serial.

Parâmetros de entrada:

- Vazio

Retorno da função:

- 1 caso a conexão tenha fechado com sucesso, 0 caso tenha falhado

DWORD

Function FechaSocket: boolean;

Função chamada para finalizar uma conexão do tipo Ethernet

Parâmetros de entrada:

- Nenhum

Retorno da função:

- “true” caso a conexão seja fechada com sucesso, “false ” caso falhe

BOOLEAN

Function LeAbastecimento:abast;

Ler o abastecimento atual em memória e passar o ponteiro de leitura de abastecimentos para o próximo;

Parâmetros de entrada:

- Nenhum

Retorno da função:

- abast: Retorna os dados do abastecimento, como preço total, volume abastecido, número do bico, etc. em uma estrutura definida na DLL. (veja a definição do tipo “abast” no final desse documento).

ABAST

Function LeAbastecimentoPAF1(autoIncrement:boolean=true):AbastPAF1;

Utilizado para ler os abastecimentos com identificação de cliente e frentista, este comando quando não passado nenhum parâmetro de entrada, executa automaticamente o comando de incremento.

Parâmetros de entrada:

- Nenhum ou false para não auto incrementar

Retorno da função:

- | | |
|---|-----------|
| <ul style="list-style-type: none"> AbastPAF1 :Retorna os dados do abastecimento, como preço total, volume abastecido, número do bico, informação de identfid. em uma estrutura definida na DLL. (veja a definição do tipo “abast” no final desse documento). | ABASTPAF1 |
|---|-----------|

Procedure Incrementa;

Função que tem por finalidade informar ao concentrador que o abastecimento atual já foi lido e armazenado.

Parâmetros de entrada:

- | | |
|--|--|
| <ul style="list-style-type: none"> Nenhum | |
|--|--|

Retorno da função:

- | | |
|--|--|
| <ul style="list-style-type: none"> Nenhum | |
|--|--|

Function LeVisualizacao():OnLine;

Comando para visualizar os abastecimentos em andamento.

Parâmetros de entrada:

- | | |
|--|--|
| <ul style="list-style-type: none"> Nenhum | |
|--|--|

Retorno da função:

- | | |
|--|--------|
| <ul style="list-style-type: none"> Online: Estrutura composta de 48 posições, contendo código de bico e o volume que o mesmo está abastecendo. As posições não utilizadas retornarão no campo “bico”, o valor “00”. (veja a definição do tipo “OnLine” no final desse documento). | Online |
|--|--------|

Function LeStatus():multistatus;

Comando utilizado para ler a situação de cada bomba conectada ao equipamento.

Parâmetros de entrada:

- | | |
|--|--|
| <ul style="list-style-type: none"> Nenhum | |
|--|--|

Retorno da função:

- Multistatus: Estrutura de dados de 48 posições, que contem o atual estado das bombas conectadas ao concentrador. (veja a definição do tipo “multistatus” no final desse documento).

MULTISTATUS

Function AutoLibera (bico:string): Error;

Libera bomba para abastecimentos.

Parâmetros de entrada:

- bico: código de bico que deseja liberar a bomba

STRING

Retorno da função:

- Error: é um tipo de dados definido pelo usuário, “None”, significa que o comando foi executado corretamente; (veja a definição do tipo ERROR no final desse documento).

ERROR

Function AutorizaAbast (bico:string): Error;

Autoriza bomba para realizar apenas um abastecimento. (funcional quando a bomba foi colocada anteriormente em modo Bloqueio, após o final do abastecimento a bomba retorna ao status Bloqueada).

Parâmetros de entrada:

- bico: código de bico que deseja liberar a bomba

STRING

Retorno da função:

- Error: é um tipo de dados definido pelo usuário, “None”, significa que o comando foi executado corretamente; (veja a definição do tipo ERROR no final desse documento).

ERROR

Function BloqueiaBico (bico:string): Error;

Bloqueia bomba para abastecimentos

Parâmetros de entrada:

- bico: código de bico que deseja liberar a bomba

STRING

Retorno da função:

- | | |
|---|-------|
| <ul style="list-style-type: none"> Error: é um tipo de dados definido pelo usuário, "None", significa que o comando foi executado corretamente; (veja a definição do tipo ERROR no final desse documento). | ERROR |
|---|-------|

Function ParaBomba (bico:string): Error;

Parar Abastecimento (não implementado em todas as bombas)

Parâmetros de entrada:

- | | |
|---|--------|
| <ul style="list-style-type: none"> bico: código de bico que deseja liberar a bomba | STRING |
|---|--------|

Retorno da função:

- | | |
|---|-------|
| <ul style="list-style-type: none"> Error: é um tipo de dados definido pelo usuário, "None", significa que o comando foi executado corretamente; (veja a definição do tipo ERROR no final desse documento). | ERROR |
|---|-------|

Function AlteraPreco (bico:string; preco:double; decimais:byte): error;

Comando utilizado para alterar o preço da bomba pela automação. (o valor será alterado no display somente quando iniciado um novo abastecimento)

Parâmetros de entrada:

- | | |
|---|--------------------------|
| <ul style="list-style-type: none"> Bico: código de bico para alterar o preço Preço: O preço a ser alterado Decimais: Quantas casas depois da virgula | STRING
DOUBLE
BYTE |
|---|--------------------------|

Retorno da função:

- | | |
|---|-------|
| <ul style="list-style-type: none"> Error: é um tipo de dados definido pelo usuário, "None", significa que o comando foi executado corretamente; (veja a definição do tipo ERROR no final desse documento). | ERROR |
|---|-------|

Function Preset (bico:string; valor:double): Error;

Comando utilizado para determinar o valor do abastecimento.

Parâmetros de entrada:

- Bico: código de bico
- Valor: valor a ser predeterminado

STRING
DOUBLE

Retorno da função:

- Error: é um tipo de dados definido pelo usuário, "None", significa que o comando foi executado corretamente; (veja a definição do tipo ERROR no final desse documento).

ERRO

Function ConsultaEncerrante (modo:char; bico:string): Encerrante;

Retorna o encerrante para o bico passado como parâmetro, (aconselhado utilizar o comando de leitura de abastecimento para acompanhar os encerrantes)

Parâmetros de entrada:

- bico : Código de bico;
- modo: Modo do retorno do encerrante (\$: Valor; L: Litros; N: Numero de série; U: PPL;P: Ponteiro da memória de abastecimentos;)

STRING

CHAR

Retorno da função:

Encerrante : É uma estrutura definida na DLL Companytec, esta, possui os campos "bico", onde retornará o código do bico lido, e "valor", que retornará o valor proveniente do pedido, em dinheiro ou volume.

ENCERRANTE

Function getclock:string;

Retorna o relógio do concentrador

Parâmetros de entrada:

- Nenhum

Retorno da função:

- Data no formato: DD/MM/AA HH:MM

STRING

Function SetClock(par:string):boolean;

Altera o relógio do concentrador

Parâmetros de entrada:

- par: 'AUTO' para o concentrador sincronizar com o relógio do computador

STRING

Retorno da função:

- boolean: true, caso consiga sincronizar, false caso falhe

BOOLEAN

Function GetVisualizacaoId:shortstring;

Utilizado para ler o andamento de um abastecimento com o respectivo código IdentFid que o autorizou

Parâmetros de entrada:

- Nenhum

Retorno da função:

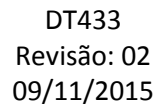
- Shortstring: código de bico mais o código do identificador para cada abastecimento em andamento

SHORTSTRING

Function FidIdent:IFid;

Caso o identificador lido pelo sensor não esteja cadastrado na memória do IdentFid II, essa informação será enviada ao PC (quando o mesmo solicitar), que por sua vez, autorizará ou não a bomba a abastecer, mediante envio de comando para a automação.

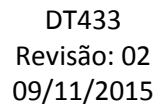
Parâmetros de entrada:

[illegible]

<ul style="list-style-type: none"> ○ 1 Reservado ○ 2 Libera Bombas ○ 4 Respeita turnos ○ 8 Liga máquina de lavagem ○ 6 Libera bombas / Respeita turnos ○ A Libera bombas / Liga máquina de lavar <ul style="list-style-type: none"> • tag: Código do Identificador • t1in:turno inicial A;(hhmm) • t1out:turno final A;(hhmm) • t2in:turno inicial B;(hhmm) • t2out:turno final B;(hhmm) 	<p>STRING</p> <p>STRING</p> <p>STRING</p> <p>STRING</p> <p>STRING</p>
Retorno da função:	
<ul style="list-style-type: none"> • INTEGER: posição onde foi armazenado o identificador, ou 0 em caso de falha 	<p>INTEGER</p>

Function deleteTagFid (posicao:integer; tag:string): integer;	
Função que deleta um identificador, dado a posição e a tag de identificação	
Parâmetros de entrada:	
<ul style="list-style-type: none"> • Posição: posição na memória onde se encontra o identificador • Tag: código de identificação do cartão a ser excluído 	<p>INTEGER</p> <p>STRING</p>
Retorno da função:	
<ul style="list-style-type: none"> • Integer: posição deletada ou 0 em caso de falha 	<p>INTEGER</p>

procedure clearTagFid;	
Função para limpar a lista de cartões de identificação	
Parâmetros de entrada:	
<ul style="list-style-type: none"> • Nenhum 	



- Nenhum

Faz a leitura dos identificadores armazenados na memória do CBC-06.

- Nro: Posição na memória

INTEGER

- o comando retorna resposta como descrito no protocolo de comunicação CBC

SHORTSTRING

Gerencia os cartões colocando e removendo estes da lista negra, é possível também limpar a lista de cartões da lista negra.(É recomendado utilizar o comando de lista negra em vez de excluir um cartão da memória)

- modo:
 - 'l': libera da lista negra
 - 'b': coloca na lista negra
 - 'c': limpa lista negra

STRING

- TAG: código de identificador para ser colocado na lista

STRING

- True em caso de sucesso , false caso contrario

BOOLEAN

Function PresetId (bico,tag:string;

cliente,libera:boolean;

valpreset,tempo:integer): Boolean;

Executa o commando de preset com a identificação do cartão de identificação, com o poder de liberar a bomba ou não.

Parâmetros de entrada:

• Bico: código de bico	STRING
• Tag: código de identificação	STRING
• Cliente: flag para configurar o preset como de cliente ou de frentista	BOOLEAN
• Liberar: flag para liberar a bomba para abastecimento	BOOLEAN
• Valpreset: valor do preset	INTEGER
• Tempo: tempo para que comece o abastecimento ou que o frentista passe o cartão	INTEGER

Retorno da função:

• True caso o preset ocorra com sucesso, false em caso de erro	BOOLEAN
--	---------

Funções compatíveis com VB6

As funções compatíveis com VB6 tem o formato VB_nomedafunção. Estas funções podem ser utilizadas pelos usuários de C# passando os parâmetros como referencia.

Sistemas que antes foram implementados com VB6 e que agora migraram para o VB.NET perceberão que algumas funções não vão funcionar como o desejado, para solucionar este problema, deve-se utilizar as funções compatíveis com .NET.

Function VB_OpenSerial(np: integer):integer;

Função para abrir uma conexão do tipo serial com o concentrador

Parâmetros de entrada:

- | | |
|--|---------|
| • np : numero da porta para ser aberta a conexão | INTEGER |
|--|---------|

Retorno da função:

- | | |
|---|---------|
| • retorna "1" em caso de sucesso e "0" caso contrário | INTEGER |
|---|---------|

Function VB_OpenSocket(ip: string):integer;

Função para abrir uma conexão do tipo TCP/IP com o concentrador

Parâmetros de entrada:

- | | |
|--|--------|
| • ip : número do ip para fazer a conexão | STRING |
|--|--------|

Retorno da função:

- | | |
|---|---------|
| • retorna "1" em caso de sucesso e "0" caso contrário | INTEGER |
|---|---------|

Function VB_CloseSerial:integer;

Função para fechar uma conexão serial

Parâmetros de entrada:

- | | |
|---------|--|
| • Vazio | |
|---------|--|

Retorno da função:

- | | |
|---|---------|
| • Retorna "1" em caso de sucesso, "0" caso contrário. | INTEGER |
|---|---------|

Function VB_CloseSocket:integer;

Função para fechar uma conexão TCP/IP

Parâmetros de entrada:

- | | |
|---------|--|
| • Vazio | |
|---------|--|

Retorno da função:

- Retorna "1" em caso de sucesso, "0" caso contrário.

INTEGER

Function VB_ReadSale(var st:string):integer;

Faz a leitura de um abastecimento.

Parâmetros de entrada:

- st: string por referencia que é passada vazia e retorna o abastecimento no formato do protocolo de comunicação CBC

STRING

Retorno da função:

- Retorna "1" em caso de sucesso, "0" caso contrário.

INTEGER

Function VB_ReadSalePAF(var st:string):integer;

Faz a leitura de um abastecimento do tipo PAF, caso o sistema não esteja utilizando Identifid os campos de identificador devem ser desprezados.

Parâmetros de entrada:

- st: string por referencia que é passada vazia e retorna o abastecimento no formato do protocolo de comunicação CBC

STRING

Retorno da função:

- Retorna "1" em caso de sucesso, "0" caso contrário.

INTEGER

function VB_NextSale:integer;

Função para realizar o incremento do ponteiro de abastecimento;

Parâmetros de entrada:

- Vazio

Retorno da função:

- Desprezável

INTEGER

function VB_ReadState(var st:string):integer;

Função que retorna o status do endereço (lado de bomba), com cada caractere representando um endereço.

Parâmetros de entrada:

- St: string passada por referencia que retorna o estado de cada lado da bomba no formato do protocolo de comunicação CBC

STRING

Retorno da função:

- Retorna "1" em caso de sucesso, "0" caso contrário.

INTEGER

Function VB_AutPump(bico:string):integer;

Função para autorizar um abastecimento e após o abastecimento voltar para o estado anterior.

Parâmetros de entrada:

- Código de bico

STRING

Retorno da função:

- Retorna "1" em caso de sucesso, "0" em caso contrário.

INTEGER

Function VB_BlockPump(bico:string):integer;

Função para bloquear um lado de bomba.

Parâmetros de entrada:

- Código de bico

STRING

Retorno da função:

- Retorna “1” em caso de sucesso, “0” caso contrário.

INTEGER

Function VB_FreePump(bico:string):integer;

Função para mudar o modo de operação para livre.

Parâmetros de entrada:

- Código de bico

STRING

Retorno da função:

- “1” em caso de sucesso, “0” caso contrário.

INTEGER

Function VB_StopPump(bico: String):integer;

Função que para um abastecimento em progresso, ou quando o endereço está com modo pronto;

Parâmetros de entrada:

- Código de bico

STRING

Retorno da função:

- “1” em caso de sucesso, “0” caso contrário.

INTEGER

function VB_PresetPump(bico,cash:string):integer;

Função que predetermina um valor para realizar o abastecimento, o abastecimento irá parar quando alcançar o valor passado como parâmetro.

O estado do endereço irá para o modo pronto após a execução do comando.

Parâmetros de entrada:

• Bico : código de bico para fazer a predeterminação	STRING
• Cash: valor em dinheiro com 6 caracteres (12,99 = "001299")	STRING
Retorno da função:	
• "1" em caso de sucesso, "0" caso contrário.	INTEGER

function VB_SetPrice(bico,price:pchar):integer;	
Função que pausa alterar o preço unitário do produto em venda no bico. Somente será visível a alteração quando um novo abastecimento for realizado.	
Parâmetros de entrada:	
• bico: código de bico para alterar o preço	PCHAR
• preço: valor a ser alterado o preço	PCHAR
Retorno da função:	
• "1" em caso de sucesso, "0" caso contrário.	INTEGER

function VB_ReadTotalsCash(bico:pchar):integer;	
Função para leitura de encerrante em valor.	
Parâmetros de entrada:	
• bico: código de bico para consultar encerrante	PCHAR
Retorno da função:	
• Em caso de sucesso, retorna o total em dinheiro, caso contrário retorna "-1"	INTEGER

function VB_ReadTotalsVolume(bico:string):integer;	
Função para leitura de encerrante em volume	

Parâmetros de entrada:

- | | |
|--|--------|
| • bico: código de bico para consultar encerrante | STRING |
|--|--------|

Retorno da função:

- | | |
|--|---------|
| • Em caso de sucesso, retorna o total em dinheiro, caso contrário retorna "-1" | INTEGER |
|--|---------|

function VB_SendReceiveText(var st:string;timeout:integer):integer;

Função para fazer a comunicação utilizando um comando do protocolo.

Parâmetros de entrada:

- | | |
|---|---------|
| • st: comando no formato do protocolo de comunicação CBC por referencia | STRING |
| • timeout: tempo para esperar a resposta do concentrador | INTEGER |

Retorno da função:

- | | |
|--|---------|
| • Retorna o tamanho da resposta que estará em st ou "0" em caso de falha | INTEGER |
|--|---------|

Funções Compatíveis com C/C++/C#/VB.NET

As funções compatíveis com C/C++, C# e VB.NET são utilizadas no exemplo de comunicação disponível no Kit de desenvolvimento.

O tipo string retornado na DLL é composto de uma sequência de caracteres com o caractere '\0' no final, indicando final de string, este tipo é compatível com o tipo char*.

Quando utilizar as funções do deste capítulo para C# e VB.NET, siga o método abaixo:

```
//declaração na DLL
Function C_GetSalePAF():pchar; stdcall;

//declaração
[DllImport("companytec.dll")]
public static extern System.IntPtr C_GetSalePAF() ;

//chamada no código
IntPtr data = C_GetSalePAF();
string str = Marshal.PtrToStringAnsi(data);
```

function C_OpenSerial(np:integer):integer;

Função para abrir uma conexão do tipo serial com o concentrador

Parâmetros de entrada:

- | | |
|--|---------|
| • np : numero da porta para ser aberta a conexão | INTEGER |
|--|---------|

Retorno da função:

- | | |
|---|---------|
| • retorna "1" em caso de sucesso e "0" caso contrário | INTEGER |
|---|---------|

function C_OpenSocket(ip:pchar):integer;

Função para abrir uma conexão do tipo TCP/IP com o concentrador

Parâmetros de entrada:

- | | |
|--|-------|
| • ip : número do ip para fazer a conexão | PCHAR |
|--|-------|

Retorno da função:

- | | |
|---|---------|
| • retorna "1" em caso de sucesso e "0" caso contrário | INTEGER |
|---|---------|

function C_CloseSerial:integer;

Função para fechar uma conexão serial

Parâmetros de entrada:

- Vazio

Retorno da função:

- Retorna "1" em caso de sucesso, "0" caso contrario

INTEGER

Function C_CloseSocket:integer;

Função para fechar uma conexão TCP/IP

Parâmetros de entrada:

- Vazio

Retorno da função:

- Retorna "1" em caso de sucesso, "0" caso contrario

INTEGER

Function C_GetSalePAF():pchar;

Faz a leitura de um abastecimento do tipo PAF, caso o sistema não esteja utilizando Identifid os campos de identificador devem ser desprezados.

Parâmetros de entrada:

- Vazio

Retorno da função:

- O retorno da função é igual à função de leitura de abastecimento PAF presente no protocolo de comunicação CBC

PCHAR

Procedure C_NextSale;

Função para realizar o incremento do ponteiro de abastecimento;

Parâmetros de entrada:

- Vazio

Retorno da função:

- Vazio

function C_readState():pchar;

Função que retorna o status do endereço (lado de bomba), com cada caractere representando um endereço.

Parâmetros de entrada:

- Vazio

Retorno da função:

- O retorno está se da no mesmo formato da leitura de status presente no protocolo de comunicação CBC

PCHAR

function C_AutoPump(bico:pchar):integer;

Função para autorizar um abastecimento e após o abastecimento voltar para o estado anterior.

Parâmetros de entrada:

- Código de bico

PCHAR

Retorno da função:

- Retorna "1" em caso de sucesso, "0" em caso contrario

INTEGER

function C_BlockPump(bico:pchar):integer;

Função para bloquear um lado de bomba.

Parâmetros de entrada:

- | | |
|------------------|-------|
| • Código de bico | PCHAR |
|------------------|-------|

Retorno da função:

- | | |
|---|---------|
| • Retorna "1" em caso de sucesso, "0" caso contrário. | INTEGER |
|---|---------|

function C_FreePump(bico:pchar):integer;

Função para mudar o modo de operação para livre.

Parâmetros de entrada:

- | | |
|------------------|-------|
| • Código de bico | PCHAR |
|------------------|-------|

Retorno da função:

- | | |
|---|---------|
| • "1" em caso de sucesso, "0" caso contrário. | INTEGER |
|---|---------|

function C_StopPump(bico:pchar):pchar;

Função que para um abastecimento em progresso, ou quando o endereço está com modo pronto;

Parâmetros de entrada:

- | | |
|------------------|-------|
| • Código de bico | PCHAR |
|------------------|-------|

Retorno da função:

- | | |
|---|---------|
| • "1" em caso de sucesso, "0" caso contrário. | INTEGER |
|---|---------|

function C_PresetPump(bico,cash:pchar):integer;

Função que predetermina um valor para realizar o abastecimento, o abastecimento irá parar quando alcançar o valor passado como parâmetro.

O estado do endereço irá para o modo pronto após a execução do comando.

Parâmetros de entrada:

- | | |
|---|-------|
| • Bico : código de bico para fazer a predeterminação | PCHAR |
| • Cash: valor em dinheiro com 6 caracteres (12,99 = "001299") | PCHAR |

Retorno da função:

- | | |
|---|---------|
| • "1" em caso de sucesso, "0" caso contrário. | INTEGER |
|---|---------|

Function C_SetPrice(bico,preco:pchar):integer;

Função que pausa alterar o preço unitário do produto em venda no bico. Somente será visível a alteração quando um novo abastecimento for realizado.

Parâmetros de entrada:

- | | |
|---|-------|
| • bico: código de bico para alterar o preço | PCHAR |
| • preço: valor a ser alterado o preço | PCHAR |

Retorno da função:

- | | |
|---|---------|
| • "1" em caso de sucesso, "0" caso contrário. | INTEGER |
|---|---------|

function C_ReadTotalsCash(bico:pchar):integer;

Função para leitura de encerrante em valor.

Parâmetros de entrada:

- | | |
|--|-------|
| • bico: código de bico para consultar encerrante | PCHAR |
|--|-------|

Retorno da função:

- | | |
|---|---------|
| • Em caso de sucesso, retorna o total em dinheiro, caso contrário retorna menor que 0 | INTEGER |
|---|---------|

function C_ReadTotalsVolume(bico:pchar):integer;

Função para leitura de encerrante em volume

Parâmetros de entrada:

- | | |
|--|-------|
| • bico: código de bico para consultar encerrante | PCHAR |
|--|-------|

Retorno da função:

- | | |
|---|---------|
| • Em caso de sucesso, retorna o total em dinheiro, caso contrário retorna valor menor que "0" | INTEGER |
|---|---------|

Function C_Visualize:pchar;

Função que retorna para cada bico abastecendo o código de bico mais o valor em tempo real do abastecimento. Esta função não deve ser utilizada para calculo, pois a mesma apresenta atraso nos dados e imprecisão.

Parâmetros de entrada:

- | | |
|---------|--|
| • Vazio | |
|---------|--|

Retorno da função:

- | | |
|--|-------|
| • Retorna para cada abastecimento ocorrendo, o código de bico mais o valor atual | PCHAR |
|--|-------|

Function C_GetClock:pchar;

Função que retorna o relógio do concentrador

Parâmetros de entrada:

- | | |
|---------|--|
| • Vazio | |
|---------|--|

Retorno da função:

- | | |
|--|-------|
| • Retorna o relógio no formato DD/MM/AA[ESPAÇO]HH:MM | PCHAR |
|--|-------|

Function C_SetClock(par:pchar):boolean;

Função para configurar o relógio do concentrador

Parâmetros de entrada:

- | | |
|--|-------|
| <ul style="list-style-type: none"> “AUTO” para que o concentrador sincronize com o horário do computador, ou uma string no formato DDHHMM | PCHAR |
|--|-------|

Retorno da função:

- | | |
|--|---------|
| <ul style="list-style-type: none"> Retorna true em caso sucesso, false caso contrário | BOOLEAN |
|--|---------|

Function C_SendReceiveText(comando:shortstring):shortstring;

Função para fazer a comunicação utilizando um comando do protocolo.

Parâmetros de entrada:

- | | |
|---|-------------|
| <ul style="list-style-type: none"> Comando: contem o comando no formato do protocolo CBC | SHORTSTRING |
|---|-------------|

Retorno da função:

- | | |
|---|-------------|
| <ul style="list-style-type: none"> Retorna o uma string no formato do protocolo de comunicação CBC | SHORTSTRING |
|---|-------------|

Estruturas de dados

Abast

- value :boolean;
- total_dinheiro :currency;
- total_litros :double;
- PU :currency;
- tempo :string[8];
- canal :string[2];
- data :string[10];
- hora :string[5];
- st_full :string[55];
- registro :integer;
- encerrante :real;
- integridade :boolean;
- checksum :boolean;

AbastPAF1

- value :boolean;
- total_dinheiro :currency;
- total_litros :double;
- PU :currency;
- tempo :string[8];
- codbico :string[2];
- numbico :integer;
- numtanque :integer;
- voltanque :integer;
- codcombustivel :integer;
- seriecbc :integer;
- tipocbc :char;
- data :string[10];
- hora :string[5];
- st_full :string[123];
- registro :integer;
- encerrantel :double;

- **encerranteF** :double;
- **integridade** :boolean;
- **checksum** :boolean;
- **tag1** :string[16];
- **tag2** :string[16];

OnLine

- **Litragem** :array [1..48] of real;
- **Bico** :array [1..48] of string[2];

MultiStatus

- **Status** :array[1 .. 48] of StOptions;

Encerrante

- **Bico** :string[2];
- **Valor** :real;



Companytec Automação e Controle Ltda.

Av. Ferreira Viana, 1421 - Areal - 96080-000 - Pelotas - RS

www.companytec.com.br

Fone: (53) 3284-8100

suporte@companytec.com.br