



# **FILIZOLA**

## **PcScale**

DLL 32 Bits para leitura de balanças

Versão da DLL : 4.0.1.1

Versão do Manual : 2.05

## Índice

1) Introdução.....	
2) Licença para uso da DLL (hardkey).....	
3) Funções Básicas.....	
3.1) Funções básicas de configuração da balança.....	
3.1.1) Função ConfiguraBalanca.....	
3.1.2) Função SalvaParametrosBalanca.....	
3.1.3) Função ObtemParametrosBalanca.....	
3.1.4) Função ObtemNomeBalanca.....	
3.2) Funções básicas de leitura da balança.....	
3.2.1) Função InicializaBalanca.....	
3.2.2) Função ObtemInformacao.....	
3.3) Função básica de envio de preço/kg a balança CS.....	
3.3.1) Função EnviaPrecoCS.....	
3.4) Função básica de exibição das mensagens de erro.....	
3.4.1) Função ExibeMsgError.....	
4) Funções Avançadas.....	
4.1) Funções avançadas de manipulação da porta serial.....	
4.1.1) Funções avançadas de abertura e configuração da porta serial.....	
4.1.2) Função avançada de fechamento da porta serial.....	
4.2) Função para leitura das balanças IDS, ID1000, IQPLUS 810, 9091, 8132, BP, MF, LINHA E IDC e TRU-TEST-SR2000.....	
4.3) Função para leitura e escrita da balança CS.....	
4.3.1) Função para leitura da balança CS.....	
4.3.2) Função para escrita na balança CS.....	
4.4) Função de leitura dos indicadores ID-U e ID-M.....	
4.5) Função de leitura da balança CI.....	

## **1) Introdução**

A DLL PcScale possui funções que permitem leitura do peso dos indicadores / balanças mais comuns no mercado.

Ela pode ser usada por qualquer linguagem capaz de carregar uma DLL em ambiente Windows 32 bits. (Ver observação para o NT/2000/XP no tópico 2).

A leitura do peso das balanças pode ser através de comunicação serial ou Ethernet.

Quando serial, a leitura se dá através da conexão das balanças a uma das portas de comunicação (COM1, COM2, COM3 ou COM4) de um computador padrão IBM-PC. Não é necessária placa adicional no computador.

Quando Ethernet, por meio de um acessório (opcional) instalado na balança, a conexão é feita a rede Ethernet, onde a balança terá um IP e Porta que a indentifique.

### **Observação Importante!!!**

A maioria dos computadores normais utilizam-se da interrupção quatro (4) para as COMs 1 e 3 e da interrupção três (3) para as COMs 2 e 4.

Dado isto se deve atentar para não utilização do mouse na porta que utilize a mesma interrupção de uma balança, o que causará o não funcionamento do mouse e da leitura da balança.

Isto também faz com que não seja possível fazer a leitura simultânea de balança que utilizem a mesma interrupção.

As funções da DLL podem ser divididas em dois conjuntos: Funções Avançadas e Funções Básicas.

As Funções Avançadas manipulam diretamente a serial e seus parâmetros requerem um conhecimento maior do programador para seu uso.

As Funções Básicas (recomendamos) utilizam as funções avançadas e oferecem ao programador uma forma simplificada de leitura da balança inclusive com suporte de telas para configuração dos parâmetros.

No pacote PcScale, no diretório “Win32\DLL”, além da DLL, encontra-se uma pasta com exemplos de utilização nas linguagens mais comum de mercado.

**Dica Importante:** Nas pastas “Exemplos” encontram-se aplicativos executáveis “exe” completos, que permitem configurar os parâmetros da serial e efetuar a leitura de todos os modelos de balança. A primeira coisa a ser feita é usar um destes aplicativos para testar se o cabo serial que interliga a balança ao computador está funcionando e se as configurações da balança estão corretas, para depois então testar o seu software desenvolvido.

## 2) Licença para uso da DLL (hardkey)

Para alguns modelos de indicador/balança o uso da DLL é condicionado a uma licença que é representada por uma Hardkey, que deve ser adquirida no departamento de vendas industrial (0800 –17 – 8077).

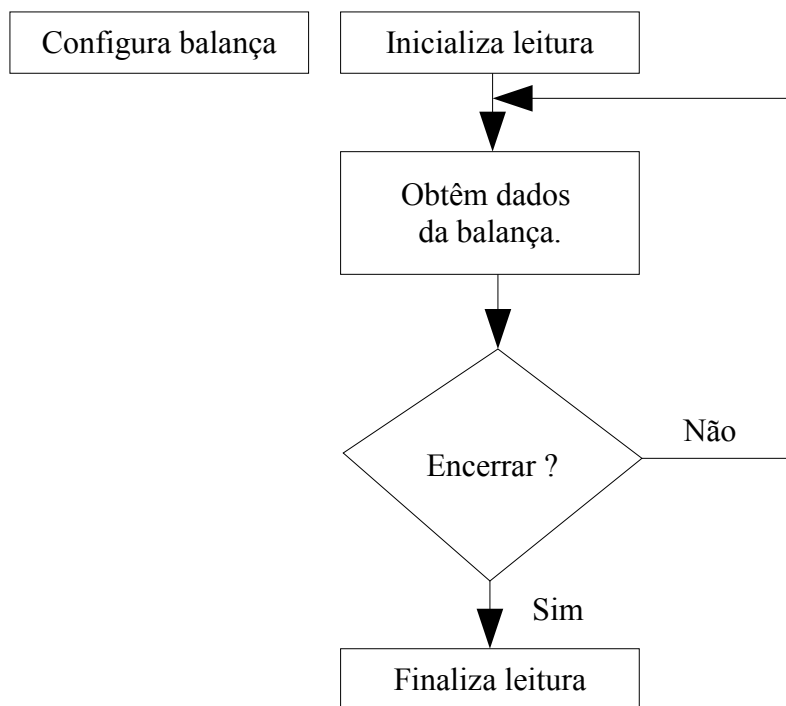
Para verificar a lista de Indicadores/Balanças contidas na DLL PcScale e detalhes sobre a licença de software, abra o arquivo C:\Filizola\PcScale\Licenca.pdf.

**Windows NT/2000/XP:** Para que o PcScale reconheça a hardkey, após sua instalação é necessário reiniciar o computador e executar o programa Setup.exe do diretório C:\Filizola\PcScale\Proteq.

## 3) Funções Básicas

As funções básicas da DLL são divididas em dois conjuntos: Um conjunto se destina a configuração dos parâmetros da balança e o outro se destina a obtenção dos dados provenientes das balanças.

O fluxograma abaixo demonstra a seqüência lógica de utilização das funções:



### 3.1) Funções básicas de configuração da balança

#### 3.1.1) Função ConfiguraBalanca

A primeira coisa a ser feita é a configuração dos parâmetros da balança.

A função ConfiguraBalanca abre a tela abaixo que servirá de interface com o operador para configuração dos parâmetros necessários a balança. Após a escolha dos parâmetros, ao clicar em “Salvar” as informações são salvas em um arquivo (ScaleParsX.sys: onde X é o número da balança) na pasta da aplicação onde foi chamada a DLL. As demais funções da DLL farão uso desta configuração para leitura da balança:



#### Boolean ConfiguraBalanca(int Balanca, HNWD Aplicativo)

##### Retorno:

Verdadeiro (true) se a configuração foi realizada ou

Falso (false) caso a configuração não tenha sido realizada ou o parâmetro balança tenha sido informado com um valor fora do range 0 a 7;

##### Parâmetros:

**Balança:** Valor inteiro (entre 0 e 7) que representa o número da balança que se deseja configurar. As funções básicas permitem trabalhar com até 8 balanças simultaneamente.

**Aplicativo:** Valor que representa o ponteiro do Aplicativo (handle) que esta chamando a função. Todas as aplicações Windows 32 possuem um Handle.

**A configuração da balança necessita ser feita uma única vez** e só precisa ser invocada novamente caso haja necessidade de se alterar os parâmetros. Após a alteração dos parâmetros de uma balança, caso a balança esteja com o processo de leitura inicializado, será necessário finaliza-lo

em seguida inicializa-lo novamente para que as alterações tenham efeito.

### **3.1.2) Função SalvaParametrosBalanca**

Caso o programador **não deseje fazer uso da tela de configuração**, então ele poderá salvar manualmente as configurações da balança através da função SalvaParametrosBalanca. Com o advento da comunicação Ethernet foi criado a função SalvaParametrosBalancaEthernet, que é semelhante a anterior porém com parâmetros necessários ao novo padrão. A função anterior foi mantida para compatibilizar a nova DLL com os programas que foram escritas com ela anteriormente:

**Boolean SalvaParametrosBalanca(int Balança, int Modelo, int Porta, int BaudRate)**  
**Boolean SalvaParametrosBalanca(int Balança, int Modelo, int Porta, int BaudRate, int Tipo, char \* IP, int Port)**

**Retorno:**

Verdadeiro (true) se os parâmetros foram salvos com sucesso ou  
Falso (false) caso haja houve falha durante a gravação do arquivo das configurações ou o parâmetro balança tenha sido informado com um valor fora do range 0 a 7;

**Parâmetros:**

**Balança:** Valor inteiro (entre 0 e 7) que representa o número da balança que se deseja salvar os parâmetros.

**Modelo:** Valor inteiro que representa o modelo do indicador/balança o qual se deseja ler, sendo:

- 0 = Indicador Filizola modelo IDS com protocolo contínuo.
- 1 = Indicador Rice-Lake modelo IQ Plus 810.
- 2 = Indicador Toledo modelo 9091.
- 3 = Indicador Toledo modelo 8132.
- 4 = Indicador Filizola modelo IDS com protocolo demanda ou indicador ID10000.

**Observação importante:** O IDS sai de fabrica configurado com o protocolo demanda.

- 5 = Balança Filizola Linha BP.
- 6 = Balança Filizola Linha MF.
- 7 = Indicador Filizola modelo IDC.
- 8 = Balança Filizola Linha E.
- 9 = Balança Filizola Linha CS ou Pluris.
- 10 = Indicador Filizola modelo IDM.
- 11 = Indicador Filizola modelo IDU.
- 12 = Indicador Tru-Test modelo SR2000.
- 13 = Balança Filizola Linha CI

**Usuários de linguagem “C”:** No arquivo PcScale.h existem definições para cada um dos indicadores/balanças (BIDS, BIQP810, B9091, B8132, BID10000, BBP, BMF, BIDC, BE,

BTRU\_TEST\_SR2000, BCI, BPLURIS).

**Porta:** Valor inteiro que representa a porta serial a qual será conectada a balança, podendo variar de 1 a 8 representando respectivamente COM1 a COM8.

**BaudRate:** Valor inteiro que representa a velocidade de comunicação que esta programada a balança, podendo ser (1200, 2400, 4800, 9600, 19200, 38400). Verificar nos documentos “Guia de Comunicação” do modelo da balança para obter informações de velocidade padrão e como configurar as balanças.

**Tipo:** Valor inteiro que representa o tipo de conexão da balança, sendo 0 (zero) Serial e 1 (um) Ethernet.

**IP:** Conjunto de caracteres que representam o número IP da balança na rede Ethernet. Ex.: 192.168.0.30

**Port:** Valor inteiro que representa o número da Porta de conexão Ethernet. geralmente 1001 ou 9001.

**Observação Importante:** As maiorias das balanças/indicadores contidas no PcScale possuem apenas a velocidade de comunicação configurável. Os demais parâmetros (DataBits, StopBits, Parity) são fixos. As balanças/indicadores que permitem configurar todos os parâmetros (Rice-Lake IQPlus 810 e Tru-Test SR2000) devem ter seus parâmetros configurados para:

DataBits: 8

StopBits: 1

Parity : None.

### 3.1.3) Função ObtemParametrosBalanca

Além das funções acima descritas, a Dll oferece duas outras funções auxiliares, que permitem obter os parâmetros salvos nos arquivos para por exemplo exibi-los na tela. Para leitura da balança não é necessário o uso destas funções.

Função ObtemParametrosBalanca para obtenção dos parâmetros configurados para uma determinada balança. Com o advento da comunicação Ethernet foi criado a função ObtemParametrosBalancaEthernet, que é semelhante a anterior porém com parâmetros necessários ao novo padrão. A função anterior foi mantida para compatibilizar a nova DLL com os programas que foram escritas com ela anteriormente:

**Boolean ObtemParametrosBalanca(int Balança, int Modelo, int Porta, int BaudRate)**  
**Boolean ObtemParametrosBalanca(int Balança, int Modelo, int Porta, int BaudRate, int Tipo, char \*IP, int Port)**

**Retorno:**

Verdadeiro (true) se existir a configuração para a balança especificada ou  
Falso (false) se não existir a configuração para a balança especificada ou o parâmetro balança tenha sido informado com um valor fora do range 0 a 7.

**Parâmetros:**

**Balança:** Valor inteiro que representa a balança a qual se deseja obter os parâmetros configurados.

**Modelo:** Variável inteira onde será retornado o modelo da balança/indicador configurado para a balança especificada.

**Porta:** Variável inteira onde será retornada a porta serial configurada para conexão da balança.

**BaudRate:** Variável inteira onde será retornada a velocidade de comunicação configurada para balança especificada.

**Tipo:** Variável inteira onde será retornada o tipo de conexão da balança, sendo 0 (zero) Serial e 1 (um) Ethernet.

**IP:** Variável Array de caracteres (tamanho 17) onde será retornado o número IP configurado para a balança na rede Ethernet. Ex.: 192.168.0.30

**Port:** Variável inteira onde será retornado o número da Porta de conexão Ethernet configurado para a balança.

### 3.1.4) Função ObtemNomeBalanca

Função para obtenção do nome do modelo da balança:



**Void ObtemNomeBalanca(int Modelo, char \* Nome)**

**Retorno:** Nenhum.

**Parâmetros:**

**Modelo:** Valor inteiro representativo do modelo ao qual se deseja obter o nome. (veja os valores possíveis para o modelo na função SalvarParametros).

**Nome:** Variável string onde será retornado o nome da balança representativa do modelo especificado. A variável “Nome” deve necessariamente ter no mínimo 50 caracteres.

## **3.2) Funções básicas de leitura da balança**

Após a configuração da balança deve-se iniciar a tarefa de leitura da balança.

### **3.2.1) Função InicializaBalanca**

A função InicializaBalanca obtém do arquivo os dados configurados para a balança especificada, programa a porta serial e ativa uma tarefa de leitura constante da balança.

**Boolean InicializaBalanca(int balança)**

**Retorno:**

Verdadeiro (true): caso a inicialização da leitura da balança tenha sido realizada com sucesso ou Falso (false): caso a inicialização da balança não tenha sido realizada com sucesso. Neste caso o sistema gravará uma mensagem com o motivo da falha que poderá ser exibida através da função ExibeMsgErro.

As mensagens mais comuns são:

**“O número da balança deve estar entre 0 e 7.”:** Exibido quando o parâmetro Balança for informado com um valor que não está entre 0 e 7.

**“A leitura desta balança já está inicializada. Use FinalizaLeitura(n) para parar a leitura.”:** Exibido quando se tenta inicializar a leitura de uma balança que já está inicializada.

**“Os parâmetros da balança não foram configurados.”:** Exibido quando se tenta inicializar uma balança cujos parâmetros não foram configurados.

**“Erro abrindo porta de comunicação especificada.”:** Exibido quando a porta de comunicação configurada não existir no computador.

**“Porta de comunicação inválida.” :** Exibido quando a porta de comunicação configurada é um número inválido.

**“Acesso a porta de comunicação não permitido.”:** Exibido quando é configurado um porta de leitura para balança que já está sendo usada por outro dispositivo, como por exemplo pelo o mouse.

**Parâmetros:**

**Balança:** Valor inteiro representativo da balança que se deseja inicializar.

### 3.2.2) Função ObtemInformacao

Uma vez inicializada a leitura da balança, para obter uma informação da balança, basta chamar a função ObtemInformacao:

**double ObtemInformacao(int Balança, int Informação)**

**Retorno:**

Valor inteiro ou decimal representativo da informação solicitada.

A função poderá retornar o valor -99999 nos seguintes casos:

- A) Quando o valor do parâmetro balança informado não estiver entre 0 e 7.
- B) Quando o valor do parâmetro informação não estiver entre 0 e 11.
- C) Quando a informação solicitada não existir para o modelo de indicador/balança.

Obs.: Nos casos A e B o sistema gravará uma mensagem que poderá ser exibida com a função ExibeMsgErro.

**Parâmetros:**

**Balança:** Valor inteiro (entre 0 e 7) representativo da balança que se deseja obter a informação.

**Informação:** Valor inteiro (entre 0 e 11) representativo da informação que se deseja obter, sendo:

0	Retorna o status da balança. Valores de retorno possíveis: 0: Balança em erro. 1: Leitura da balança com sucesso, mas o peso esta variando. 2: Leitura da balança com sucesso e peso estável. 3: Balança fora de range. (sobrecarga ou alívio de prato). 4: Licença de software (hardkey) não encontrada.  <b>Usuários de linguagem "C":</b> No arquivo PcScale.h existem definições para cada um dos retornos de status (dfnBalancaErro, dfnPesoOscilando, dfnPesoEstavel, dfnBalancaForaRange, dfnLicencaNaoEncontrada).
1	Retorna o valor de peso bruto constante na balança. (*)
2	Retorna o valor de peso tara constante na balança. (*)
3	Retorna o valor de peso líquido constante na balança. (*)
4	Retorna o valor do contador de pesagens. Somente para os indicadores Filizola modelos IDU e IDM.
5	Retorna o código digitado na balança. Somente para os indicadores Filizola modelos IDU e IDM e a balança Filizola modelo CS.
6	Valor monetário representativo do preço por quilo constante na balança. Somente para a balança Filizola modelo CS.
7	Valor monetário representativo do preço total (preço/kg x peso liquido) constante na balança. Somente para a balança Filizola modelo CS.
8	Número de casas decimais visualizadas no display da balança. As balanças BP e CS não transmitem o ponto decimal.

9	Peso médio por peça em gramas (somente para balança CI)
10	Data de fabricação digitada na balança no formato ddmmaa (somente balança CI)
11	Data de validade digitada na balança no formato ddmmaa (somente balança CI)

(\*) **Observações sobre os valores de peso:** Os valores são retornados com símbolo decimal (quando houver) e formatação de acordo com a configuração de “Número” constante em “Configurações Regionais” do Windows. Por serem numéricos, os zeros não significativos a esquerda da para inteira e os zeros não significativos a direita da parte fracionária são suprimidos, fazendo com que o tamanho da informação e número de casas decimais variem.

### 3.3) Função básica de envio de preço/kg a balança CS

Dentro do universo das funções básicas também se encontra uma função para envio do preço por quilo para a balança CS:

#### 3.3.1) Função EnviaPrecoCS

**Boolean EnviaPrecoCS(int balanca, doublé Preço)**

**Retorno:**

Verdadeiro (true) quando o envio do preço/kg ocorreu com sucesso (recebimento confirmado pela balança) ou

Falso (false) quando a balança não estiver inicializada, o modelo da balança configurado não for a balança CS, a balança estiver desligada ou desconectada ou o parâmetro balança tenha sido informado com um valor fora do range 0 a 7.

**Parâmetros:**

**Balança:** Valor inteiro que representa a balança a qual se deseja enviar o preço por quilo.

**Preço:** Valor fracionário representativo do preço por quilo que se deseja enviar. O valor deve ser maior que 0 e menor que 9999,99 e deve ter no máximo duas casas decimais.

### 3.4) Função básica de exibição das mensagens de erro

Todas as funções com retorno gravam internamente uma mensagem de detalhes na ocorrência de um erro. Para exibir esta mensagem o programador deve chamar a função ExibeMsgErro imediatamente após a chamada da função que gerou o erro. Ex.:

```
If( ! InicializaLeitura(0) ) ExibeMsgErro(this->Handle);
```

#### 3.4.1) Função ExibeMsgErro

**void ExibeMsgErro(HNWD Aplicativo)**

**Retorno:** Nenhum.

**Parâmetros:**

**Aplicativo:** Valor que representa o ponteiro do Aplicativo (handle) que esta chamando a função. Todas as aplicações Windows 32 possuem um Handle.

**3.4.2) Função ObtemMsgErro**

O programador poderá usar esta função caso tenha necessidade de copiar a mensagem de erro para uma variável própria.

**void ExibeMsgErro(char \*mensagem)**

**Retorno:** Nenhum.

**Parâmetros:**

**mensagem:** Valor que representa o ponteiro para um buffer de caracteres onde será copiada a última mensagem de erro. Deve ser devinido com tamanho de 101 caracteres.

**4) Funções Avançadas**

Obs.: De preferência ao uso da funções básicas. As funções avançadas são um método antigo de leitura da balança e foram mantidas por compatibilidade.

As funções avançadas da DLL são divididas em dois conjuntos: Um conjunto se destina a manipulação da porta serial (abrir, configurar e fechar a porta serial) e outro a leitura da balança.

Para todos os modelos de balança contemplados pela DLL as funções de manipulação da serial são as mesmas, apenas para a leitura da balança a função varia para alguns modelos.

A DLL permite a leitura simultânea das balanças conectadas ao computador.

**4.1) Funções avançadas de manipulação da porta serial**

São duas as funções de manipulação da serial. Uma para abertura e configuração dos parâmetros de comunicação da porta serial do computador de acordo com os parâmetros exigidos/configurados na balança e outra para fechamento da porta serial do computador.

**4.1.1) Funções avançadas de abertura e configuração da porta serial**

Efetua a abertura e programação dos parâmetros de comunicação da porta serial especificada.

**int AttachScale(HANDLE \*hScale, int ComPort, int BaudRate, int ByteSize, int StopBits, int Parity)**

**Onde:**

**HANDLE \*hScale** – Ponteiro retornado pela função para o dispositivo de comunicação especificado.

**int ComPort** – Número da porta de comunicação onde esta conectada a balança para leitura ( 1, 2, 3 e 4 para COM1, COM2, COM3, COM4 ).

**int BaudRate** – Velocidade de comunicação entre o computador e a balança ( 300, 600, 1200, 2400, 4800, 9600, 19200, 36600).

**Obs.:** Alguns computadores antigos podem não apresentar velocidades acima de 9600 bauds.

**int ByteSize** - Número de bits por byte ( 4 a 8 ).

**int StopBits** – Número de bits de término ( 0 = 1 Stopbits; 1 = 1,5 StopBits; 2 = 2 StopBits ).

**Obs.:** É um erro comum na utilização do AttachScale o programador passar o valor 1 quando deseja expressar 1 StopBit, erroneamente, pois conforme mostrado acima representa 0 representa 1 StopBits. Quando isto acontece a função AttachScale retorna o erro “Parâmetros de Comunicação Inválidos”.

**int Parity** – Paridade ( 0 = sem paridade, 1 = paridade impar, 2 = paridade par )

A função AttachScale retorna o status da operação de abertura do dispositivo de comunicação, que pode ser:

- 0 = Dispositivo aberto com sucesso.
- 2 = Erro abrindo dispositivo
- 3 = Dispositivo inexistente
- 4 = Muitos dispositivos/arquivos abertos
- 5 = Acesso ao dispositivo não permitido. Dispositivo já aberto ou em uso por outro programa
- 6 = Parâmetros de comunicação inválidos.
- 7 = Erro programando timeout de comunicação.
- 31 = Erro Genérico
- 100 = Arquivo vazio
- 101 = Não conseguiu abrir devido a falta de espaço em disco.
- 106 = Recebendo caracteres inválidos.

#### **4.1.2) Função avançada de fechamento da porta serial**

Fecha o dispositivo de comunicação apontado pelo \*hScale.

**void DettachScale(HANDLE \*hScale)**

O AttachScale já contém no início de seu código um DettachScale que permite sua chamada para reprogramação do dispositivo de comunicação sem a chamada do DettachScale manualmente.

Em balanças onde o protocolo é contínuo (o frame é enviado o tempo todo: IDS, IQPLUS, 9091,

8132) é altamente recomendado abrir a porta de comunicação com AttachScale instantes antes do processo de leitura da balança e após a conclusão da leitura fechar imediatamente a porta de comunicação com o DettachScale. Isto se deve pelo fato da porta serial para o Windows ser trata como um arquivo, assim que se abre a porta todo byte recebido é colocado neste arquivo, e quando deixamos este arquivo aberto sem leitura ele cresce indeterminadamente, e no momento que formos efetuar a leitura, leremos um informação que chegou a muito tempo atrás, dando a impressão que o valor de peso lido é diferente do valor indicado na balança.

#### **4.2) Função para leitura das balanças IDS, ID1000, IQPLUS 810, 9091, 8132, BP, MF, LINHA E IDC e TRU-TEST-SR2000.**

Efetua a leitura da balança especificada no canal de leitura apontado pelo hScale, obtido na chamada da função AttachScale.

**int ReadScale(HANDLE \*hScale, int Indicator, char \*gross, char \*tare, char \*net, char \*unit)**

**Onde:**

**HANDLE \*hScale** – Ponteiro para a porta de comunicação programada pelo AttachScale.

**Indicator** – Modelo do indicador ou balança conectado ao canal de leitura.

- 0 = Indicador Filizola modelo IDS com protocolo contínuo.
- 1 = Indicador Rice-Lake modelo IQ Plus 810.
- 2 = Indicador Toledo modelo 9091.
- 3 = Indicador Toledo modelo 8132.
- 4 = Indicador Filizola modelo IDS com protocolo demanda ou indicador ID10000.

**Observação importante:** O IDS sai de fabrica configurado com o protocolo demanda.

- 5 = Balança Filizola Linha BP.
- 6 = Balança Filizola Linha MF.
- 7 = Indicador Filizola modelo IDC.
- 8 = Balança Filizola Linha E.
- 9 = Balança Filizola Linha CS.
- 10 = Indicador Filizola modelo IDM.
- 11 = Indicador Filizola modelo IDU.
- 12 = Indicador Tru-Test modelo SR2000.
- 13 = Balança Filizola Linha CI

No arquivo PcScale.h existem definições para cada um dos indicadores/balanças (BIDS, BIQP810, B9091, B8132, BID10000, BBP, BMF, BIDC, BE, BTRU\_TEST\_SR2000, BCI).

A Filizola se põe a disposição para consulta de inclusão de modelos de balanças que não estão compreendidas pelo PcScale.

**char \*gross** – Ponteiro para buffer onde será retornado o valor de peso bruto.

**char \*tare** – Ponteiro para buffer onde será retornado o valor de tara.

**char \*net** – Ponteiro para buffer onde será retornado o valor de peso líquido.

**char \*unit** – Ponteiro para buffer onde será retornado a unidade programada na balança.

Os buffers gross, tare e net devem ser definidos com 9 caracteres. O buffer unit deve ser definido com 3 caracteres.

Os valores retornados nos buffers são strings terminadas por nulo (ultimo caracter = 0x0h) onde os oito primeiros caracteres contem os pesos preenchidos com zeros não significativos, com o ponto decimal (quando existir e for informado pela balança ) e em caso de pesos negativos o primeiro caracter conterá o sinal '-' (0x2Dh).

A função ReadScale retorna o status da operação definidos no PcScale.h como:

**ScaleError (0x0h)** – Erro de leitura da balança. Não foi possível efetuar a comunicação entre o computador e a balança. Neste caso os buffers gross, tare, net e unit são completamente preenchidos por nulos.

**ScaleInMotion (0x1h)** – Peso lido não estável (variando).

**ScaleStable (0x02h)** – Peso lido estável.

**ScaleOutOfRange(0x3h)** – Sobrecarga ou alívio do prato. Peso sobre a plataforma da balança acima ou abaixo do máximo permitido.

**ProteqNotFound (0x4h)** – Chave de licença do software não encontrada.

### **4.3) Função para leitura e escrita da balança CS.**

A balança CS, por apresentar algumas características diferentes das demais, possui uma função específica para leitura do peso e escrita do Preço/kg.

#### **4.3.1) Função para leitura da balança CS.**

Efetua a leitura da balança CS no canal de leitura apontado pelo hScale, obtido na chamada da função AttachScale.

**int ReadCS(HANDLE \*hScale, char \*peso, char \*vunit, char \*vtotal, char \*codigo)**

**Onde:**

**HANDLE \*hScale** – Ponteiro para a porta de comunicação programada pelo AttachScale.



**char \*peso** – Ponteiro para buffer onde será retornado o valor de peso.

**char \*vunit** – Ponteiro para buffer onde será retornado o valor de Preço/kg constante na balança.

**char \*vtotal** – Ponteiro para buffer onde será retornado o valor de Preço Total constante na balança (Peso x Preço/kg).

**char \*codigo** – Ponteiro para buffer onde será retornado o código do produto constante na balança.

Os buffers peso, vunit e vtotal devem ser definidos com 7 caracteres. O buffer código deve ser definido com 5 caracteres.

Os valores retornados nos buffers são strings terminadas por nulo (ultimo caracter = 0x0h) onde os seis primeiros caracteres contem os respectivos valores preenchidos com zeros não significativos, **sem** o ponto decimal e em caso de valores negativos o primeiro caracter conterá o sinal '-' (0x2Dh).

A função ReadScale retorna o status da operação definidos no PcScale.h como:

**ScaleError (0x0h)** – Erro de leitura da balança. Não foi possível efetuar a comunicação entre o computador e a balança. Neste caso os buffers peso, vunit, vtotal e código são completamente preenchidos por nulos.

**ScaleInMotion (0x1h)** – Peso lido não estável (variando).

**ScaleStable (0x02h)** – Peso lido estável.

**ScaleOutOfRange(0x3h)** – Sobrecarga ou alívio do prato. Peso sobre a plataforma da balança acima ou abaixo do máximo permitido.

**ProteqNotFound (0x4h)** – Chave de licença do software não encontrada

### **4.3.2) Função para escrita na balança CS.**

A balança CS permite receber serialmente o valor de Preço/ kg.

A função abaixo descrita efetua o envio do Preço/kg a balança.

**int WriteCS(HANDLE \*hScale, char \*vunit)**

**Onde:**

**HANDLE \*hScale** – Ponteiro para a porta de comunicação programada pelo AttachScale.

**char \*vunit** – Ponteiro para buffer que contém o valor de Preço/kg que será enviado a balança.

Os buffer vunit deve ser definido com 7 caracteres.

O valor constante no buffer deve ser uma string terminada por nulo (ultimo caracter = 0x0h) onde os seis primeiros caracteres devem conter o valor de Preço/kg preenchido com zeros não significativos e **sem** o ponto decimal.

A função WriteCS retorna o seguinte status:

**(0x0h)** – Erro no envio do preço.

**(0x1h)** – Envio efetuado com sucesso.

#### **4.4) Função de leitura dos indicadores ID-U e ID-M**

Efetua a leitura dos indicadores ID-U e ID-M no canal de leitura apontado por hScale, obtido na chamada da função AttachScale.

**int ReadIDM (HANDLE \*hScale, char far \*net, char far \*tare, char far \*cont, char far \*cod)**

**Onde:**

**HANDLE \*hScale** – Ponteiro para a porta de comunicação programada pelo AttachScale.

**char \*gross** – Ponteiro para buffer onde será retornado o valor do peso liquido

**char \*tare** – Ponteiro para buffer onde será retornado o valor da tara da balança, no caso do indicador estar programado para o envio do mesmo.

**char \*cont** – Ponteiro para buffer onde será retornado o valor da contagem de peças sobre a balança

**char \*cod** – Ponteiro para buffer onde será retornado o código do produto constante na balança.

Os buffers Gross, tare, ser definidos com 9 caracteres. Os buffers cod e cont deve ser definido com 7 e 6 caracteres, respectivamente.

Os valores retornados nos buffers são strings terminadas por nulo (ultimo caracter = 0x0h) onde os seis primeiros caracteres contem os respectivos valores preenchidos com zeros não significativos, **com** o ponto decimal e em caso de valores negativos o primeiro caracter conterà o sinal '-' (0x2Dh).

A função ReadIDM retorna o status da operação definidos no PcScale.h como:

**ScaleError (0x0h)** – Erro de leitura da balança. Não foi possível efetuar a comunicação entre o computador e a balança. Neste caso os buffers peso, vunit, vtotal e código são completamente preenchidos por nulos.

**ScaleInMotion (0x1h)** – Peso lido não estável (variando).

**ScaleStable (0x02h)** – Peso lido estável.

**ScaleOutOfRange(0x3h)** – Sobrecarga ou alívio do prato. Peso sobre a plataforma da balança acima ou abaixo do máximo permitido.

**ProteqNotFound (0x4h)** – Chave de licença do software não encontrada

#### **4.5) Função de leitura da balança CI**

Efetua a leitura da balança CI no canal de leitura apontado por hScale, obtido na chamada da função AttachScale.

**int ReadCI (HANDLE \*hScale, char far \*gross, char far \*tare, char far \*net, char far \*qtd, char far \*cod, char far \*pmed, char far \*dfab, char far \*dval)**

**Onde:**

**HANDLE \*hScale** – Ponteiro para a porta de comunicação programada pelo AttachScale.

**char \*gross** – Ponteiro para buffer onde será retornado o valor do peso bruto.

**char \*tare** – Ponteiro para buffer onde será retornado o valor da tara da balança.

**char \*net** – Ponteiro para buffer onde será retornado o valor do peso líquido.

**char \*qtd** – Ponteiro para buffer onde será retornado a quantidade de peças sobre a balança resultado da operação de contagem.

**char \*cod** – Ponteiro para buffer onde será retornado o código do produto digitado na balança.

**char \*pmed** – Ponteiro para buffer onde será retornado o valor do peso médio por peça calculado pela balança na operação de amostragem.

**char \*dfab** – Ponteiro para buffer onde será retornado o valor da data de fabricação digitada na balança.

**char \*dval** – Ponteiro para buffer onde será retornado o valor da data de validade digitada na balança.

Os buffers Gross, tare, devem ser definidos com 9 caracteres. Os buffers cod e qtd deve ser definido com 7 e 6 caracteres, respectivamente. Os buffers dfab e dval devem ser definidos com 7 caracteres.

Os valores retornados nos buffers são strings terminadas por nulo (ultimo caracter = 0x0h) onde os seis primeiros caracteres contem os respectivos valores preenchidos com zeros não significativos, com o ponto decimal e em caso de valores negativos o primeiro caracter conterà o sinal '-' (0x2Dh).

A função ReadCI retorna o status da operação definidos no PcScale.h como:

**ScaleError (0x0h)** – Erro de leitura da balança. Não foi possível efetuar a comunicação entre o computador e a balança. Neste caso os buffers peso, vunit, vtotal e código são completamente preenchidos por nulos.

**ScaleInMotion (0x1h)** – Peso lido não estável (variando).

**ScaleStable (0x02h)** – Peso lido estável.

**ScaleOutOfRange(0x3h)** – Sobrecarga ou alívio do prato. Peso sobre a plataforma da balança acima ou abaixo do máximo permitido.

**ProteqNotFound (0x4h)** – Chave de licença do software não encontrada