

## LePeso.dll

O device driver LePeso.dll foi desenvolvido para fornecer aos programadores uma forma padrão de leitura das informações indicadas nos visores dos seguintes modelos de balanças produzidas pela Urano: UDP, UDC, UDI, UD MINI, UDC MINI, UDC Premium, UDC P LIGHT, UDC 6000/1 S, UDC POP, US POP, CP POP, UR10000, UDC CO e UDC CO E.

Este device driver é compatível com o Windows 9x, 2000, NT e XP. Sua instalação consiste em copiar o arquivo "LePeso.dll" para dentro do diretório raiz do sistema operacional Windows, ou então, para dentro do subdiretório "System" deste mesmo diretório.

As funções contidas neste device driver são as seguintes:

### **\_LePeso**

Função: Responsável pela leitura das informações indicadas no visor da balança;  
Protótipo: char \*LePeso();  
Parâmetros: Não requer nenhum parâmetro.  
Retorno: Retorna um ponteiro para *string*.  
Comentário: Esta função retorna uma *string* de caracteres (terminada com um NULL) proveniente da balança. O formato desta *string* é descrito em detalhes no manual de cada modelo de balança.

Tamanho das strings para cada modelo escolhido:

**Modelo 0:** 17 caracteres.  
**Modelo 1:** 9 caracteres.  
**Modelo 2:** 46 caracteres.  
**Modelo 3:** 55 caracteres.  
**Modelo 4:** 66 caracteres.  
**Modelos 5 e 6:** 69 caracteres.

Transcorrido o tempo máximo (*timeout*) para efetuar a leitura do peso, essa *string* é preenchida com asteriscos (\*).

### **\_AbrePortaSerial**

Função: Abre a porta serial para comunicação.  
Protótipo: int AbrePortaSerial(char \*);  
Parâmetros: Nome da porta serial: "COM1", "COM2", ...  
Retorno: 1 - Sucesso ou 0 - Insucesso.  
Comentário: Esta função configura a porta serial com base no modelo de balança e modo de operação da mesma.

## **\_FechaPortaSerial**

Função: Fecha a porta serial para comunicação.  
Protótipo: int FechaPortaSerial();  
Parâmetros: Não requer nenhum parâmetro.  
Retorno: 1 - Sucesso ou 0 - Insucesso.

## **\_AlteraModeloBalanca**

Função: Altera o modelo de balança.  
Protótipo: void AlteraModeloBalanca(int);  
Parâmetros: Modelo de balança: 0, 1, 2, 3, 4, 5 ou 6;  
Retorno: Não retorna nada.  
Comentário: Veja abaixo qual parâmetro deverá ser enviado.

**Parâmetro 0:** balanças antigas UDP, UDC, UDI, UD-MINI e UDC-MINI dotadas de processador TMS-URANO11 ou anterior.

**Parâmetro 1:** balanças antigas que se utilizam do protocolo Urano6.

**Parâmetro 2:** todas as balanças que se utilizam do protocolo Urano12 - balanças UDC POP, UDC Premium, UDC P LIGHT, UDC 6000/1 S, UR10000, UDC CO e UDC CO E. Balanças antigas UDP, UDC, UDI, UDC-MINI e UD-MINI dotadas de processador TMS-URANO12.

**Parâmetro 3:** balança US POP.

**Parâmetro 4:** balança CP POP revisão L.

**Parâmetro 5:** balança CP POP a partir da revisão 1.

**Parâmetro 6:** balanças que se utilizam do protocolo URANO C. Balanças UDC P LIGHT, UDC 6000/1 S e UR10000.

Toda vez que esta função é chamada, a porta serial é fechada, devendo-se, portanto, abrí-la novamente para ser utilizada na leitura do peso.

## **\_AlteraModoOperacao**

Função: Altera o modo de operação.  
Protótipo: void AlteraModoOperacao(int);  
Parâmetros: Modo de operação: 0 ou 1;  
Retorno: Não retorna nada.  
Comentário: O modo de operação determina a forma de leitura do valor do peso, que pode ser por solicitação do computador (parâmetro = 0) ou pelo pressionamento da tecla de impressão/envia presente na balança (parâmetro = 1). O *timeout* para o modo 0 é de 2 s e para o modo 1, de 4 s. Para configurar o modo de operação na balança, por favor, consulte seu manual. Toda vez que esta função é chamada, a porta serial é fechada, devendo-se, portanto, abrí-la novamente para ser utilizada na leitura do peso.

### **EXEMPLO DE UTILIZAÇÃO:**

```
#include <iostream.h>
#include <conio.h>

typedef char *(*LEPESO)();
typedef int (*ABREPORTASERIAL)(char *);
typedef int (*FECHAPORTASERIAL)(void);
typedef void (*ALTERAMODELOBALANCA)(int);
typedef void (*ALTERAMODOOPERACAO)(int);

int main(int argc, char* argv[])
{
    HINSTANCE hinstLib;
    LEPESO ProcLePeso;
    ABREPORTASERIAL ProcAbrePortaSerial;
    FECHAPORTASERIAL ProcFechaPortaSerial;
    ALTERAMODELOBALANCA ProcAlteraModeloBalanca;
    ALTERAMODOOPERACAO ProcAlteraModoOperacao;

    hinstLib = LoadLibrary("LePeso.dll");

    if(hinstLib != NULL) {
        ProcLePeso = (LEPESO)GetProcAddress(hinstLib, "_LePeso");
        ProcAbrePortaSerial = (ABREPORTASERIAL)GetProcAddress(hinstLib, "_AbrePortaSerial");
        ProcFechaPortaSerial = (FECHAPORTASERIAL)GetProcAddress(hinstLib,
        "_FechaPortaSerial");
        ProcAlteraModeloBalanca = (ALTERAMODELOBALANCA)GetProcAddress(hinstLib,
        "_AlteraModeloBalanca");
        ProcAlteraModoOperacao = (ALTERAMODOOPERACAO)GetProcAddress(hinstLib,
        "_AlteraModoOperacao");

        (ProcAlteraModeloBalanca)(1);

        (ProcAlteraModoOperacao)(1);

        (ProcAbrePortaSerial)("COM1");

        while(!kbhit()) cout << (ProcLePeso)();

        (ProcFechaPortaSerial)();
    }

    FreeLibrary(hinstLib);

    return 0;
}
```