

## Импорт замеров с виброприборов компании “Вибро-Центр”

v1.00 17.04.2023

[AndreySchekalev@vibrocenter.ru](mailto:AndreySchekalev@vibrocenter.ru)

### Схема уровней для связи с прибором

3. Разбор замера как структуры конкретного прибора. Преобразование замера в универсальный формат замера	ViAna-2: DoImportViana2() в файле ImportViana2.pas  Диана-2М: ImportKorsarM() в файле ImportKorsar.pas	
2. Передача бинарных данных через последовательность структур TLinkList	Обмен низкоуровневыми фреймами TLinkList  TestImportForm.pas	
1. Объект «Универсальный порт связи»	TUSBLinkPort в файле LinkPortClass.pas	TEthernetLinkPort в файле LinkPortClass.pas
0. USB или Ethernet server	USB Vendor Device:  VC_USB_VID = \$0441; // vendor ID  VC_USB_PID = \$51C9; // Product ID  VCUSBHdr.pas через библиотеку lib_usb	

## 0. USB драйвер

В примере VCUSBHdr.pas обмен идёт через библиотеку LibUsb – A cross-platform user library to access USB devices.

Сайт: <https://libusb.info/>

Для Win32 можно использовать нашу прослойку VCUSB DLL через файл VCUSB DLL Hdr.pas

### ***LibUsb в Windows***

Для установки драйвера и нашего файла vusb.inf используйте:

USB/InstallDriver.exe

В каталог программы положить файл libusb-1.0.dll

### ***LibUsb в Linux***

```
sudo apt-get install libusb
```

или

```
su -c "apt-get install libusb"
```

Чтобы разрешить работу с нашим прибором по USB нужно сделать файл ~/vc.rules:

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="0441", ATTRS{idProduct}=="51c9", MODE="0666"
```

```
SUBSYSTEM=="usb_device", ATTRS{idVendor}=="0441", ATTRS{idProduct}=="51c9", MODE="0666"
```

и скопировать его:

```
sudo cp ~/vc.rules /etc/udev/rules.d/
```

```
sudo udevadm control --reload-rules
```

Пример файла лежит в

Linux/vc.rules

Используется USB Vendor Device:

```
#define USB_DEVICE_VENDOR_ID          0x0441
```

```
#define USB_DEVICE_PRODUCT_ID         0x51C9
```

```
#define USB_DEVICE_MANUFACTURE_NAME    "Vibro-Center/Dimrus"
```

```
#define USB_DEVICE_PRODUCT_NAME        "Vibro-Center/Dimrus Device"
```

Два Bulk Device endpoints:

```
#define UDI_VENDOR_EP_BULK_IN    0x81    // Передача с прибора на компьютер  
#define UDI_VENDOR_EP_BULK_OUT  0x02    // Передача с компьютера на прибор
```

Для USB 1.1 максимальный размер пакета = 64 байта

Для USB 2.0 и выше максимальный размер пакета = 512 байт

### ***Функции VCUSBHdr.pas или VCUSBDllHdr.pas:***

Возвращают S\_OK =0; или ошибки, если !=0

Инициализация – всегда вызывать один раз перед началом работы с USB. Проверяет, что драйвер LibUsb установлен.

```
function InitUsb(): HRESULT;
```

Освободить библиотеку – вызывать в конце программы

```
procedure DoneUsb();
```

Вызывать перед обменом с прибором.

Ищет наш USB device, создаёт объекты для обмена.

```
function VCUSBInit(): HRESULT;
```

Вызывать после окончания обмена – закрывает USB device, уничтожает объекты.

Можно вызывать неоднократно.

```
function VCUSBDone(): HRESULT;
```

Передать массив pBuffer размером cbBuffer байт из компьютера в прибор.

В случае успеха возвращает cbWritten – сколько байт записано.

uTimeout – таймаут на ожидание в миллисекундах. Не должен быть ==0. Например, 100

```
function VCUSBWrite(pBuffer: pointer; cbBuffer: DWORD; var cbWritten: DWORD; uTimeout: DWORD):  
HRESULT;
```

Принять cbBuffer байт в массив pBuffer размером из прибора в компьютер.

В случае успеха возвращает cbRead – сколько байт записано.

uTimeout – таймаут на ожидание в миллисекундах. Не должен быть ==0. Например, 100

```
function VCUSBRead(pBuffer: pointer; cbBuffer: DWORD; var cbRead: DWORD; uTimeout: DWORD):  
HRESULT;
```

### **Типичный сценарий обмена:**

```
InitUsb(); // В начале программы
```

```
...
```

```
If (VCUSBInit() = S_OK) then begin // Начало обмена  
    while (data_count > 0) do begin  
        VCUSBWrite() – записать команду  
        VCUSBRead() – принять ответ  
    end;  
  
    VCUSBDone(); // Конец обмена
```

```
End;
```

```
...
```

```
DoneUsb(); // В конце программы
```

## **1. Объект «Универсальный порт связи»**