

**Динамическая библиотека pv650x.dll  
для осциллографов  
PV6501A, PV6502 и PV6503.**

## Содержание.

1 Назначение.....	3
2 Описание функций.....	4
2.1 pv_List Получить список доступных устройств.....	4
2.2 pv_Open Инициализация прибора по серийному номеру.....	4
2.3 pv_Close Закрывать обмен с прибором.....	4
2.4 pv_CloseAll - Закрывать обмен со всеми устройствами.....	4
2.5 pv_PutAllReg Функция конфигурации прибора.....	4
2.6 pv_GetData Функция получения всех данных прибора.....	4
2.7 Функции работы с локальной структурой OSC.....	5
2.7.1 pv_PutOsc – функция обновления полей конфигурации осциллографа.....	5
Входные параметры, аналогичны членам структуры OSC и более подробно рассмотрены в разделе «Структура данных OSC управления осциллографом и генератором». Через указатели: att, zero, trig_lev, trig_gist передается указатели на массивы, содержащие два элемента со значениями данного параметра для каналов A1 и A2. Нулевой элемент массива соответствует каналу A1, а первый - A2. ....	5
2.7.1 pv_PutGen – функция обновления полей конфигурации генератора.....	6
2.7.2 pv_Config – Конфигурация по локальной структуре OSC.....	6
2.7.3 pv_GetAllData - Получение данных.....	6
3 Структура данных OSC управления осциллографом и генератором. ....	8
4 Структура данных служебного пакета PV65.....	13
5 Пример 1. Программа TestApplication на Borland C++.....	14
6 Пример 2. Программа PV650X_Demo на LabVIEW.....	15

## 1 Назначение.

Динамическая библиотека *pv650x.dll* предназначена для работы с осциллографами PV6501A, PV6502 и PV6503. Она включает в себя функции инициализации и закрытия обмена данными, конфигурации осциллографа, получения осциллограмм и вспомогательных данных.

Функцией *pv\_List* можно получить список серийных номеров доступных приборов. По одному из полученных серийных номеров можно открыть обмен с данными с осциллографом при помощи *pv\_Open*. Эта функция при успешном выполнении возвращает идентификатор прибора (хендл). Он используется для конфигурации, получения данных и закрытия обмена данными функцией *pv\_Close*. При использовании функции *pv\_Open* для других серийных номеров, возможна работа с несколькими приборами одновременно. Конфигурация прибора осуществляется вызовом функции *pv\_PutAllReg* или вспомогательных функций: *pv\_PutOsc*, *pv\_PutGen* и *pv\_Config*. Все параметры прибора описаны в структуре *OSC*. Указатель на нее передается при вызове функций конфигурации. Также в эту структуру возвращаются данные о текущих настройках при вызове функции получения осциллограмм *pv\_GetData* или ее обертки *pv\_GetAllData*. При получении осциллограмм помимо текущих настроек возвращаются и дополнительные данные, описанные структурой *OSCCTRL*.

## 2 Описание функций.

### 2.1 *pv\_List* Получить список доступных устройств.

*extern "C" int \_DECLSPEC \_\_stdcall pv\_List( PV\_SERIAL\_LIST strs );*

Возвращает список найденных устройств.

*Вход:*

*PV\_SERIAL\_LIST strs* – массив строк.

*Выход:*

Если меньше 0 – ошибка.

### 2.2 *pv\_Open* Инициализация прибора по серийному номеру

*extern "C" int \_DECLSPEC \_\_stdcall pv\_Open( char\* ser, DWORD\* h\_pv );*

Открывает обмен с устройством.

*Вход:*

*char\* ser* – строка с серийным номером устройства.

*DWORD\* h\_pv* – указатель на идентификатор устройства. Если инициализация успешна, в *h\_pv* сохраняется новый идентификатор.

*Выход:*

Если меньше 0 – ошибка (открыть устройство не удалось).

### 2.3 *pv\_Close* Закрывает обмен с прибором

*extern "C" int \_DECLSPEC \_\_stdcall pv\_Close( DWORD \* h\_pv );*

Закрывает обмен с устройством с идентификатором *h\_pv*.

*Вход:*

*DWORD\* h\_pv* – указатель на идентификатор закрываемого прибора

*Выход:*

Если меньше 0 – ошибка.

### 2.4 *pv\_CloseAll* - Закрывает обмен со всеми устройствами

*extern "C" void \_DECLSPEC \_\_stdcall pv\_CloseAll( void );*

Закрывает обмен со всеми открытыми устройствами.

### 2.5 *pv\_PutAllReg* Функция конфигурации прибора

*extern "C" int \_DECLSPEC \_\_stdcall pv\_PutAllReg ( DWORD\* h\_pv , POSC osc );*

Конфигурирует прибор по структуре OSC.

*Вход:*

*DWORD\* h\_pv* – указатель на идентификатор прибора.

*POSC osc* – указатель на структуру OSC с данными конфигурации.

*Выход:*

Если меньше 0 – ошибка.

### 2.6 *pv\_GetData* Функция получения всех данных прибора.

*extern "C" int \_DECLSPEC \_\_stdcall pv\_GetData ( DWORD\* h\_pv , PCHAR A1 ,  
PCHAR A2 , PCHAR D1 , PCHAR D2 , POSC osc , POSCTRL oscctrl );*

Функция получения данных о настройках и осциллограмм от прибора по его идентификатору.

*Вход:*

*DWORD\* h\_pv* – указатель на идентификатор прибора.

*P UCHAR A1, A2, D1, D2* – указатели на массивы для сохранения осциллограмм размером *MAX\_BUF\_LENGTH*. Если указатель равен *NULL* – копирование данных игнорируется.

*POSC osc* – указатель на структуру *OSC* возвращается всегда с актуальными данными.

*POSCCTRL oscctrl* – указатель на структуру *OSCCTRL* – возвращается всегда, независимо от остальных данных.

*Выход:*

Битовое поле, значение битов описано в таблице:

Значение бита	Номер бита							
	0	1	2	3	4	5	6	7
Нет данных или ошибка	0	0	0	0	0	0	0	0
Есть осциллограмма канала A1	1	x	x	x	x	x	x	x
Есть осциллограмма канала A2	x	1	x	x	x	x	x	x
Есть осциллограмма канала D1	x	x	1	x	x	x	x	x
Есть осциллограмма канала D2	x	x	x	1	x	x	x	x
Считана структура CTRL	x	x	x	x	1	x	x	x
Считана структура OSC	x	x	x	x	x	1	x	x
Ожидает завершения конфигурации	x	x	x	x	x	x	1	x
Устанавливается один раз после конфигурации	x	x	x	x	x	x	x	1

## 2.7 Функции работы с локальной структурой OSC.

Для удобной работы с библиотекой в средах LabVIEW, Matlab и т.п. для каждого открытого прибора создается структура локальная OSC. Значение ее полей можно задавать описанными ниже функциями *pv\_PutOsc* и *pv\_PutGen*. Прибор конфигурируется функцией *pv\_Config*. Текущие данные и осциллограммы можно получить функцией *pv\_GetAllData*.

### 2.7.1 pv\_PutOsc – функция обновления полей конфигурации осциллографа.

```
extern "C" int _DECLSPEC __stdcall pv_PutOsc ( DWORD* h_pv, UCHAR adc_mode, USHORT trig_mode, UCHAR trig_con, ULONG smp_per, USHORT smp_pre, USHORT smp_post, P UCHAR att, PUSHORT zero, P UCHAR trig_lev, P UCHAR trig_gist, UCHAR trig_num, UCHAR trig_slew, PULONG min_max_P, UCHAR fmetr_in, UCHAR fps );
```

*Вход:*

Входные параметры, аналогичны членам структуры OSC и более подробно рассмотрены в разделе «Структура данных OSC управления осциллографом и генератором». Через указатели: *att*, *zero*, *trig\_lev*, *trig\_gist* передается указатели на массивы, содержащие два элемента со значениями данного параметра для каналов A1 и A2. Нулевой элемент массива соответствует каналу A1, а первый - A2.

*DWORD\* h\_pv* – указатель на идентификатор прибора.

*UCHAR adc\_mode* – битовое поле управления набором данных.

*USHORT trig\_mode* – битовое поле с настройками триггера.

*UCHAR trig\_cond* – битовое поле с дополнительными условиями триггера.

*ULONG smp\_per* – период синхронизации по 10ns.

*USHORT smp\_pre* – количество предвыборок.

*USHORT smp\_post* – количество выборок после синхронизации.

*P UCHAR att* – Атенюатор канала A1 и A2.

*PUSHORT zero* – уровень нуля канала A1 и A2.

*P UCHAR trig\_lev* – уровень триггера канала A1 и A2.

*P UCHAR trig\_gist* – гистерезис триггера канала A1 и A2.

*UCHAR trig\_numb* – номер перепада, по которому срабатывает триггер.

*UCHAR trig\_slew* – скорость нарастания.

*PULONG min\_max\_P* – передается указатель на массив, в котором нулевой элемент – минимальная длительность импульса вызывающая синхронизацию, первый – максимальная.

*UCHAR fmetr\_in* – выбор входа частотомера.

*UCHAR fps* – частота обновления экрана.

*Выход:*

Если меньше 0 – ошибка.

### **2.7.1 pv\_PutGen – функция обновления полей конфигурации генератора.**

```
extern "C" int _DECLSPEC __stdcall pv_PutGen ( DWORD* h_pv, bool gen_enable, UCHAR gen_mode, UCHAR gen_mod_in, bool gen_mod_invert, UCHAR* gen_shape, double* gen_freq_span, PULONG gen_pulse, ULONG gen_mod_K, USHORT gen_ampl, PCHAR gen_mod_zero );
```

*Вход:*

*DWORD\* h\_pv* – указатель на идентификатор прибора.

*bool gen\_enable* – включение генератора.

*UCHAR gen\_mode* – режимы генератора.

*UCHAR gen\_mod\_in* – вход модулирующего сигнала.

*bool gen\_mod\_inver* – инверсия модулирующего сигнала.

*PCHAR gen\_shape* – форма сигнала основного и модулирующего генераторов.

*double\* gen\_freq\_span* – передается указатель на массив, в котором нулевой элемент частота основного генератора, первый – частота модуляции с точностью до 3.55E-5 Гц, второй – конечная частота ГКЧ с точностью 0.1 Гц.

*PULONG gen\_pulse* – передается указатель на массив, в котором нулевой элемент – период генератора импульсов, первый – длительность импульсов.

*ULONG gen\_mod\_K* – коэффициент модуляции при АМ или девиация частоты при ЧМ.

*USHORT gen\_ampl* – амплитуда генератора.

*PCHAR gen\_mod\_zero* – ноль канала A1 и A2 при внешней модуляции. Передается указатель на массив, в котором нулевой элемент ноль для канала A1, первый – для A2.

*Выход:*

Если меньше 0 – ошибка.

### **2.7.2 pv\_Config – Конфигурация по локальной структуре OSC.**

```
extern "C" int _DECLSPEC __stdcall pv_Config ( DWORD* h_pv );
```

*Вход:*

*DWORD\* h\_pv* – указатель на идентификатор прибора.

*Выход:*

Если меньше 0 – ошибка.

### **2.7.3 pv\_GetAllData - Получение данных.**

```
extern "C" int _DECLSPEC __stdcall pv_GetAllData ( DWORD* h_pv, UCHAR* A1, UCHAR* A2, UCHAR* D1, UCHAR* D2, PCHAR flags, PUSHORT addr_eq, PULONG trig_fine_pos, UCHAR*
```

*trig\_Vb\_Va\_S*, *PUSHORT trig\_P*, *double\* trig\_gen\_phase*, *double\* freq*, *USHORT\* smpl*, *UCHAR\* ver\_type\_V*);

*Вход:*

*DWORD\* h\_pv* – указатель на идентификатор прибора.

*UCHAR\* A1*, *UCHAR\* A2*, *UCHAR\* D1*, *UCHAR\* D2* – осциллограммы.

*PCHAR flags* – различные флаги. Используются младшие три бита.

Значение бита	Номер бита		
	0	1	2
Был обгон указателем чтения	1	x	x
Завершился первый проход после старта	x	1	x
Данные частотомера верны	x	x	1

*PUSHORT addr\_eq* – номер отчета, на котором произошел обгон указателем чтения указателя записи.

*PULONG trig\_fine\_pos* – уточненное положение момента синхронизации.

*UCHAR\* trig\_Vb\_Va\_S* – передается указатель на массив, в котором нулевой элемент значение кода АЦП до синхронизации, первый – после, второй – скорость нарастания импульса, вызвавшего синхронизацию.

*PUSHORT trig\_P* – измеренная длительность импульса

*double\* trig\_gen\_phase* – фаза основного и модулирующего генератора.

*double\* freq* – показания частотомера.

*USHORT\* smpl* – выборки до и после синхронизации.

*UCHAR\* ver\_type\_V* – передается указатель на массив, в котором нулевой элемент версия FPGA, первый – версия Firmware, второй – тип прибора, третий и четвертый – старший и младший байты напряжения питания платы в мВ.

### 3 Структура данных OSC управления осциллографом и генератором.

ADC\_MODE adc\_mode;

Структура управления набором данных:

```
typedef union __ADC_MODE{
unsigned char data :7;
struct
{
bool peak_detect :1; Набор отсчетов с пиковым детектором
bool select_A1:1; Выбран канал A1
bool close_in_A1 :1; Закрытый вход канала A1
bool select_A2 :1; Выбран канал A2
bool close_in_A2 :1; Закрытый вход канала A2
bool select_D1 :1; Выбран канал D1
bool select_D2 :1; Выбран канал D2
};} ADC_MODE , *PADC_MODE;
```

TRIG\_MODE trig\_mode;

Структура с настройками триггера:

```
typedef union __TRIG_MODE {
unsigned short data :10;
struct
{
bool auto_restart :1; Автоперезапуск (синхр. и предвыборки выкл.)
bool neg_sync :1; Синхронизация по срезу
bool auto_sync :1; Автоматическая синхронизация
bool sync_pt :1; Синхронизации по длительности импульса
bool sync_pt_in :1; true - Синхронизирует если длительность
импульса в заданных пределах, false - вне.
bool sync_st :1; Синхронизации по скорости нарастания
bool sync_st_less :1;
char trig_src :3; Выбора источника сигнала для триггера
};} TRIG_MODE *PTRIG_MODE;
char trig_src;
```

Выбора источника сигнала для триггера:

Триггер на канале A1	0
Триггер на канале A2	1
Триггер на канале D1	2
Триггер на канале D2	3
Триггер отключен	4

TRIG\_COND trig\_cond;

Структура с дополнительными условиями триггера:

```

typedef union __TRIG_COND unsigned char data :8;
struct
{
  bool A1_lev :1;      0 - если ноль, 1 - если единица
  bool A2_lev :1;      0 - если ноль, 1 - если единица
  bool D1_lev :1;      0 - если ноль, 1 - если единица
  bool D2_lev :1;      0 - если ноль, 1 - если единица
  bool A1_enb :1;      Включен A1
  bool A2_enb :1;      Включен A2
  bool D1_enb :1;      Включен D1
  bool D2_enb :1;      Включен D2
};}TRIG_COND, *PTRIG_COND ;

```

*ULONG* *smp1\_per*;

Период синхронизации по 10ns

*USHORT* *smp1\_pre*;

*USHORT* *smp1\_post*;

Количество выборок до и после синхронизации. Максимальное значение зависит от выбранного режима и каналов. При выборе надо исходить из того, что максимально значение выборок всего (*smp1\_pre* + *smp1\_post*):

1. Общая память 20000 отсчетов.
2. Если включен один аналоговый канал то он занимает всю память (*smp1\_pre* + *smp1\_post* = 20000). Для двух аналоговых каналов максимальное количество отсчетов 10000.
3. Включение триггерных каналов уменьшает максимальное количество отсчетов на ¼. Само значение должно быть кратно 4.
4. При включении пикового детектора максимальное количество отсчетов уменьшается в два раза.

Если устанавливается неподходящее значение *smp1\_pre* и/или *smp1\_post* библиотека автоматически корректирует его после конфигурации прибора. Скорректированные значения можно прочитать при получении данных.

*UCHAR* *att\_A1*; и *UCHAR* *att\_A2*;

Аттенюатор каналов A1 и A2.

Значение	Имя	Назначение
5	2 В/деление	аттенюатор 1:40
4	1 В/деление	аттенюатор 1:20
3	0.5 В/деление	аттенюатор 1:10
2	0.2 В/деление	аттенюатор 1:4
1	0.1 В/деление	аттенюатор 1:2
0	0.05 В/деление	аттенюатор 1:1

*UCHAR* *zero\_A1*; и *UCHAR* *zero\_A2*;

Уровень нуля каналов A1 и A2. Значение от 0 до 4095.

*UCHAR* *trig\_lev\_A1*; и *UCHAR* *trig\_lev\_A2*;

Уровень триггера каналов A1 и A2. Значение от 0 до 255.

*UCHAR* *trig\_gist\_A1*; и *UCHAR* *trig\_gist\_A2*;

Гистерезис триггера каналов A1 и A2. Значение от 0 до 255.

*UCHAR* *trig\_num*;

Номер перепада по которому срабатывает триггер. На автоматических развертках отключен (равен нулю). Значение от 0 до 255.

*UCHAR trig\_slew;*

Скорость нарастания. Значение от 0 до 255.

*ULONG min\_P u ULONG max\_P*

Минимальная и максимальная длительность импульса вызывающая синхронизация.

*UCHAR fmetr\_in*

Выбор входа частотомера:

Вход частотомера	Значение
Выключен	0
A1	1
A2	2
D1	3
D2	4

*UCHAR fps*

Частота обновления экрана в мс. Значение от 0 до 255.

Поля доступные для чтения. Запись в них игнорируется.

**При непрерывных развертках:**

*bool eq\_flag;*

Был обгон указателем чтения.

*bool end\_flag;*

Завершился первый проход после старта.

*USHORT addr\_eq;*

Номер отчета, на котором произошел обгон указателем чтения указателя записи.

*ULONG trig\_fine\_pos;*

Положение момента синхронизации в пределах периода дискретизации (10 нс).

**Дополнительные данные синхронизации:**

*UCHAR before\_V;*

*UCHAR after\_V;*

Значения аналогового отсчета до синхронизации и вызвавшего её. Значение от 0 до 255.

*USHORT trig\_P;*

Измеренная, длительность импульса вызывавшая синхронизацию.

*UCHAR trig\_S;*

Скорость нарастания импульса, вызвавшего синхронизацию. Значение от 0 до 255.

*unsigned \_\_int64 trig\_gen\_phase;*

Фаза генератора на момент срабатывания синхронизацию.

*double freq;*

Показания частотомера (Гц). Если показания частотомера не верны freq = -1.

*bool f\_valid;*

Флаг - данные частотомера верны.

*bool f\_double\_mem;*

Флаг - удвоенная память включена.

### Генератор:

*bool gen\_enable;*

Включение генератора.

*UCHAR gen\_mode;*

Режимы основного генератора.

Тип колебаний	Значение
Незатухающие	0
Генератор качающейся частоты	1
АМ	2
ФМ	3
Сумма	4
Генератор импульсов	5

*UCHAR gen\_mod\_in;*

Вход модулирующего сигнала.

Вход модулирующего сигнала	Значение
Внутренний	0x00
A1	0x10
A2	0x20
D1	0x30
D2	0x40

*bool gen\_mod\_invert;*

Инверсия модулирующего сигнала.

*UCHAR gen\_shape;*

*UCHAR gen\_mod\_shape;*

Форма сигнала основного и модулирующего генератора.

Форма волны	Значение
Синус	0
Пила 1	1
Пила 2	2
Меандр	3
Треугольные	4
Постоянный ток 4.095В	5
Постоянный ток 2.047В	6
Шум	7

*USHORT gen\_phase;*

*USHORT gen\_mod\_phase;*

Фаза основного и модулирующего генератора в момент срабатывания синхронизации осциллографа (старта развертки).

*double gen\_freq;*

Частота генератора, начальная частота при ГКЧ (Гц) с точность до 3.55E-5 Гц.

*double gen\_mod\_freq;*

Частота модуляции (Гц) с точность до 3.55E-5 Гц.

*double gen\_swip\_span;*

Конечная частота при ГКЧ (0.1 Гц).

*ULONG gen\_pulse\_T;*

Период генератора импульсов (10 нс).

*ULONG gen\_pulse\_D;*

Длительность импульсов в (10нс).

*ULONG gen\_mod\_K;*

При различных режимах модуляции: коэффициент модуляции АМ (0.1%); девиация частоты ЧМ (+-1.48939 Гц), максимальная девиация 195215.64318Гц; относительный уровень при сумме (0.1%).

*USHORT gen\_ampl;*

Амплитуда генератора (мВ). Значение 0 до 4094.

*UCHAR gen\_mod\_zero1;*

*UCHAR gen\_mod\_zero2;*

Ноль модулирующего сигнала при внешней модуляции каналов А1 и А2. Значение от 0 до 255.

## 4 Структура данных служебного пакета PV65.

Структура доступна только для чтения.

*UCHAR FPGA\_ver;*

Версия FPGA. В двоично-десятичном коде.

*UCHAR firm\_ver;*

Версия прошивки. В двоично-десятичном коде.

*ULONG serial;*

Серийный номер.

*UCHAR pv\_type;*

Тип устройства.

Тип устройства	Значение
PV6501A	0x1C
PV6502	0x19
PV6503	0x1F

*UCHAR TDC\_Rise;*

*UCHAR TDC\_Fall;*

*UCHAR TDC\_Rise2;*

*UCHAR TDC\_Fall2;*

Калибровки TDS.

*USHORT Vcc\_Scope;*

Напряжение питания платы прибора (мВ).

*USHORT Points\_Num;*

Количество снимаемых точек. Общее количество точек *smpl\_pre* + *smpl\_post*. Значение от 0 до 20000.

## 5 Пример 1. Программа TestApplication на Borland C++.

С использованием динамической библиотеки pv650x.dll написана программа TestApplication. В ней предоставлен доступ ко всем членам структуры OSC. Внешний вид программы представлен на рис. 1.

После запуска программы нажмите кнопку «Открыть» в правом нижнем углу в текстовом поле появится список доступных приборов и будет открыт первый возможный прибор. Если обмен с прибором данными открыт успешно будет распечатан идентификационный номер прибора.

По нажатию кнопки «Применить» осуществляется конфигурация прибора. Кнопка «Заккрыть» - закрывает обмен данными.

Внешний вид осциллограммы получен при конфигурации значениями по умолчанию и при соединенном выходе генератора с входом осциллографа A1.

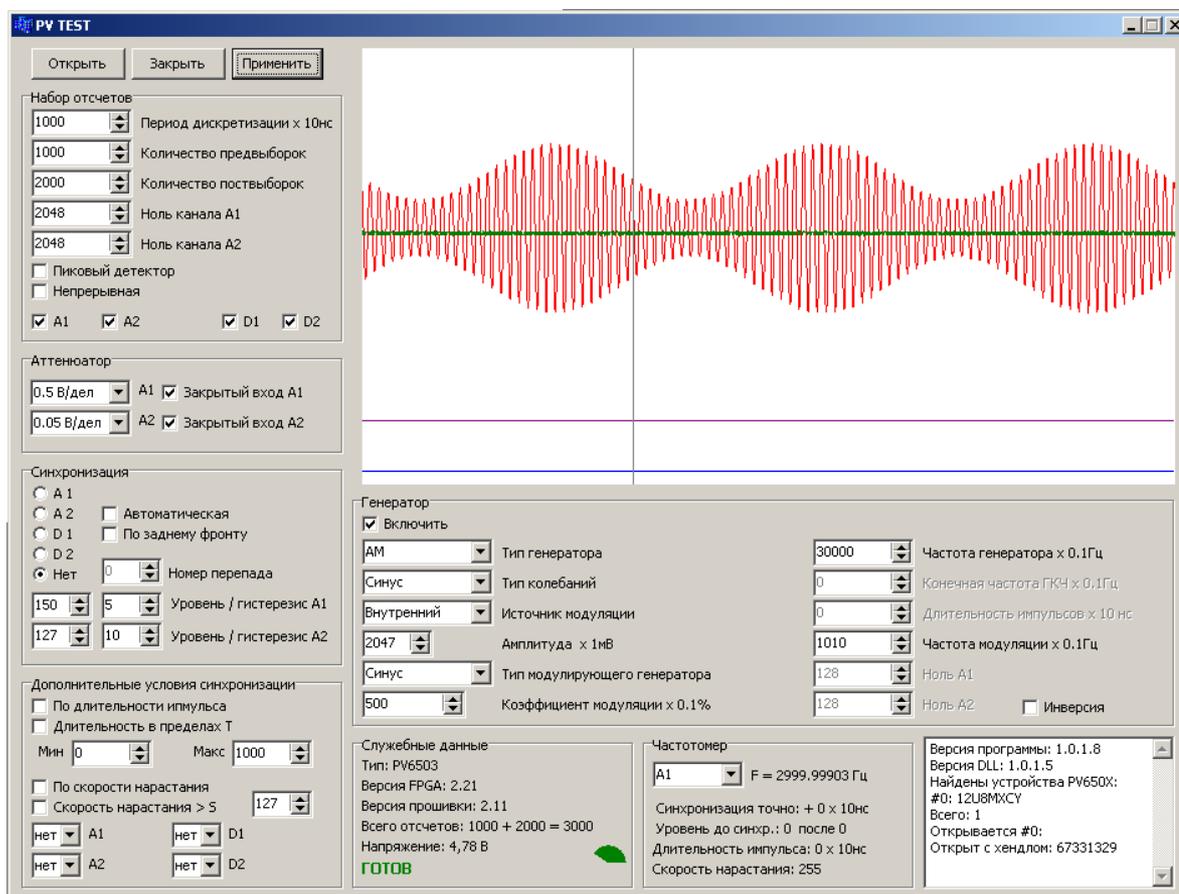


Рис. 1. Интерфейс программы TestApplication.

## 6 Пример 2. Программа PV650X\_Demo на LabVIEW.

Программа, написанная на LabVIEW, функционально повторяет программу TestApplication. Кнопка «Открыть» и «Закрыть» отсутствуют, после запуска программа сразу открывает обмен с первым доступным прибором. А по нажатию кнопки «Выход» закрывает обмен с устройством и останавливает выполнение программы. Внешний вид пользовательского интерфейса представлен на рис. 2. Блок-диаграмма на рис. 3.

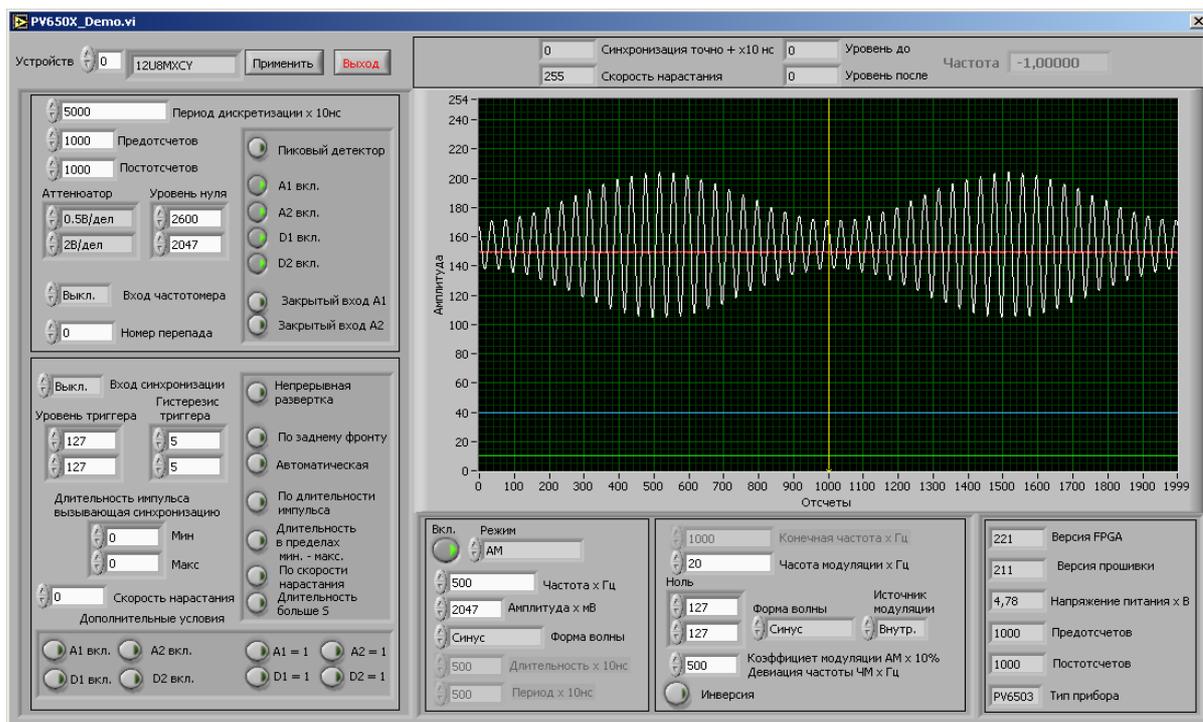


Рис. 2. Интерфейс программы PV650X\_Demo.

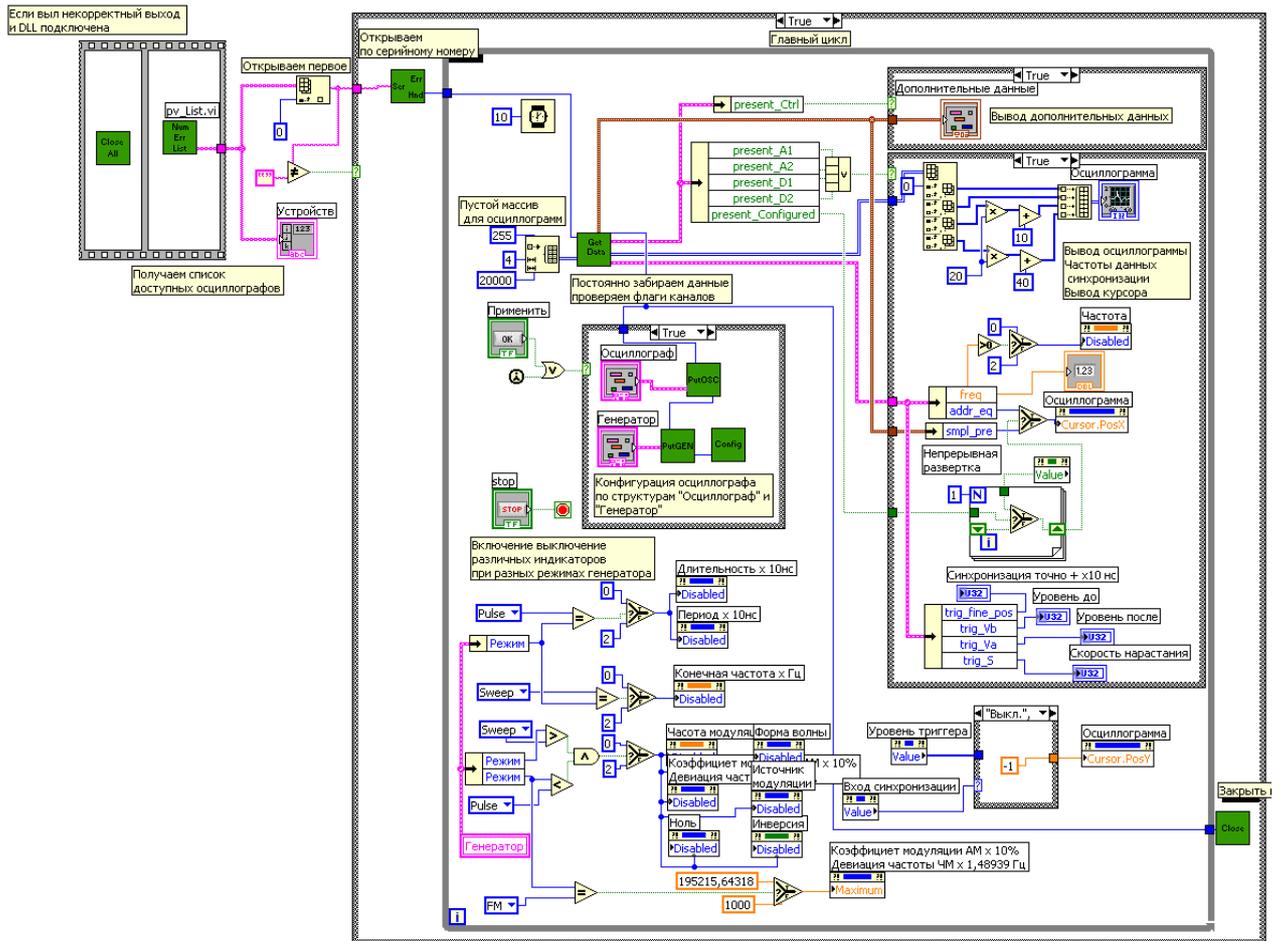


Рис. 3. Блок-диаграмма программы PV650X\_Demo.