

Устройство для регистрации и цифровой обработки аналоговых сигналов с частотой дискретизации 100 МГц ADC-DAC-125_14

АЦП 2x14 Бит, 100 МГц
ЦАП 2x14 Бит, 100 МГц
MASTER/SLAVE PCI-express

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

- Тип шины обмена данными — MASTER/SLAVE PCI-express
- Конструктивное исполнение – PCI-express
- Количество каналов АЦП — 2
- Разрядность АЦП — 14 бит
- Количество каналов ЦАП – 2
- Разрядность ЦАП – 14 бит
- Отсутствие пропуска кодов, гарантировано для АЦП — 14 бит
- Максимальная частота дискретизации для АЦП 100 МГц
- Входное сопротивление аналогового входа для АЦП – 50 ± 1 Ом
- Диапазон входного сигнала для АЦП не более ± 0.5 В
- Среднеквадратичное значение апертурного дребезга для АЦП при температуре +25С, ns (типовое значение) 1**
- Интегральная нелинейность преобразования для АЦП $\pm 2.5^* \text{ EMP}$ (тип.)
- Дифференциальная нелинейность преобразования для АЦП $\pm 0.75^* \text{ EMP}$ (тип.)
- Вход внешнего запуска
- Интегральная нелинейность преобразования для ЦАП $\pm 3,0^* \text{ EMP}$ (тип)
- Дифференциальная нелинейность преобразования для ЦАП $\pm 2,5^* \text{ EMP}$ (тип)
- Дифференциальная нелинейность преобразования для ЦАП компаратора $\pm 1,0 \text{ EMP}$ (max)
- Максимальный объем буферного ЗУ — 16 М отсчетов сигнала на канал
- Потребляемая мощность не более 10 Вт
- Напряжения питания — 5 В
- Питание устройства осуществляется через разъем AMP171826-4 (5 В)***
- Плата позволяет устанавливать перепрограммируемые вентилярные матрицы фирмы XILINX семейства Spartan-3 и Spartan-3E
- Программирование схемы цифрового автомата осуществляется от микросхемы Flash SPROM, установленной на плате
- Схема цифрового автомата обработки сигнала может быть изменена по требованию заказчика
- Плата поддерживает работу PCI интерфейса в режиме "Master/Slave" (DMA, средняя скорость передачи данных 100 MB/Sec)

* - параметры контролируются косвенными методами;

** - значение для микросхемы АЦП. Для устройства в целом параметр зависит от применяемого осциллятора который может быть установлен по ТЗ заказчика.

*** - разъем такой же как и те, что использовались для питания флоппи дисководов.

Общее описание платы

Устройство ADC-DAC-125_14 предназначено для преобразования аналоговых сигналов в цифровые коды, преобразования цифровых кодов в аналоговые сигналы, хранения этих кодов и передачи их по шине PCI-express.

Аналоговая часть устройства собрана на базе микросхемы АЦП фирмы Texas Instruments. Вход с трансформаторной развязкой, имеет входное сопротивление 50 Ом и рассчитан на работу с источником сигнала, имеющим выходное сопротивление равное 50 Ом. В случае отсутствия сигнала на входе (работа в режиме холостого хода) на выходе АЦП присутствует небольшой шум. Максимальная тактовая частота АЦП составляет 100 МГц (частота дискретизации задается коэффициентом деления тактовой частоты от 1 до 256). Результаты преобразования записываются в буферную память устройства ADC-DAC-125_14. Плата имеет два аналоговых канала.

Цифровая часть устройства собрана на базе микросхем ЦАП фирмы Texas Instruments. На плате установлены два типа преобразователей – быстрый ЦАП позволяющий работать на частоте 100 МГц и медленный, позволяющий работать на частоте 20 МГц. На быстрых ЦАП-ах построены два канала преобразования цифрового сигнала в аналоговый. Медленный ЦАП формирует сигнал, который поступает на компаратор.

Использование ПЛИС семейства Spartan-3 и Spartan-3E фирмы XILINX позволяет перепрограммировать устройство ADC-DAC-125_14 для реализации обработки аналоговых и цифровых сигналов по алгоритмам заказчика. Типовая конфигурация платы – осциллоскоп.

Процесс дискретизации запускается по команде с компьютера или по сигналу внешнего запуска и продолжается до заполнения буферной памяти. Общий объем буферной памяти 64 МБ (4 микросхемы по 16 МБ). Обмен данными между платой и персональным компьютером выполняется через DMA буферы ПК со средней скоростью 100 МБ/сек (работа PCI-express интерфейса в режиме "Master/Slave"). Центральный процессор ПК не принимает участие в передаче данных из буферной памяти платы в DMA буфер и в это время может выполнять другие задачи. По окончании процесса дискретизации плата выставляет сигнал аппаратного прерывания, который обрабатывается программой пользователя.

Плата формирует сигнал на разрешение внешнего запуска и сигнал «временного окна», которые могут использоваться для синхронного запуска внешнего процесса.

В комплект поставки входят:

- плата ADC-DAC-125_14;
- комплект разъемов для подключения к плате аналоговых сигналов;
- драйвер устройства для операционных систем Windows 2000\XP;
- тестовое программное обеспечение WIN_DO_125AS5_Oscilloscope_01 (исходный текст проекта для Delphi 7.0);
- DLL библиотека API функций для использования платы в проектах пользователя;
- руководство пользователя.

Установка и использование платы

После установки платы в свободный разъем и загрузки компьютера операционная система обнаружит устройство. На запрос о местонахождении драйвера необходимо указать путь к файлу "MDMAI.inf". Данный файл входит в комплект ПО платы. При установке драйвера под операционными системами Windows 2000\XP, в случае появления диалогового окна "Цифровая подпись не найдена" на запрос "Продолжить установку?" необходимо ответить "Да".

Поставляемый с платой драйвер, входящий в состав программного обеспечения, обеспечивает работу устройства под управлением операционных систем Windows 9x\ME\2000\XP.

После установки платы в свободный PCI-Express слот компьютера и включения ПК,

BIOS обнаружит новое устройство и идентифицирует его как "**сопроцессор**". В списке установленных устройств должна появиться запись: **Vendor ID = 0x2004** и **Device ID = 0x680C** (на многих компьютерах таблица с установленными устройствами не появляется, а сразу начинается загрузка операционной системы).

После загрузки, операционная система обнаружит в системе новое устройство "**PCI Co-processor CPU (Сопроцессор)**" и предложит установить для него драйвер. Выберите пункт "**установка из указанного места**". Далее укажите "**Поиск наиболее подходящего драйвера**" (с пометкой на дискете A:\), если ПО поставляемое с платой находится на дискете, иначе укажите полный путь к директории в которой находится драйвер "**MDMAI.sys**" и файл "**MDMAI.inf**".

При установке драйвера вручную, в окне с предложением поиска наилучшего драйвера выберите "**Установить с диска...**" и выберите каталог в котором находится драйвер "**MDMAI.sys**" и файл "**MDMAI.inf**". Далее продолжайте установку в соответствии с подсказками системы. После того, как драйвер будет установлен, перезагрузите компьютер.

После установки драйвера рекомендуется перейти в диспетчер устройств («**Свойства системы**» - «**Оборудование**» - «**Диспетчер устройств**») для того, что бы убедиться в том, что устройство и драйвер работают нормально (класс устройств **XDSPDEVICES**).

Если при запуске операционной системы устройство не было обнаружено, то либо оно неисправно, либо драйвера под него уже устанавливалось ранее в систему, и у менеджера устройств уже есть соответствующая запись. В случае, если драйвера уже устанавливались, откройте менеджер устройств, разверните класс устройств «**XDSPDEVICES**» и выберите запись отмеченную как «**ADCDAC125_14**». В свойствах этой записи надо выбрать закладку «**Драйвер**» и нажать кнопку «**Обновить...**». Далее следовать указаниям по установке драйвера.

Для проверки работоспособности платы можно воспользоваться программой **WIN_DO_125AS5_Oscilloscope**, входящей в комплект программного обеспечения, поставляемого с платой.

Описание тестовой программы цифрового осциллографа WIN_DO_125AS5_Oscilloscope

Программа цифрового осциллографа "**WIN_DO_125AS5_Oscilloscope**" предназначена для тестирования работоспособности платы ADC-DAC-125_14 и демонстрации ее основных функций. Программа поставляется с исходным текстом проекта для среды программирования Delphi.

Программный продукт распространяется "AS IS" и служит для демонстрации основных возможностей и механизмов работы с платой. Поставщик не несет ответственности за некорректность работы программы и ущерб принесенный в следствии ее возможного зависания. Пользователь может дорабатывать и модифицировать программу для использования в собственном проекте.

Программа предоставляет пользователю следующие основные возможности:

- Просмотр оцифрованного сигнала по банкам (размер банка может изменяться произвольно);
- Просмотр 2 каналов (каждый канал по отдельности, 2 канала вместе и разницу между 1-ым и 2-ым каналами);
- Однократная и циклическая дискретизация;
- Автомасштабирование сигнала, наложение сигнала, программная синхронизация;
- Вычисление глобальных (на всем массиве данных) и локальных (в пределах выбранного банка) минимума и максимума;
- Просмотр значения сигнала в точке;

- Диагностика процесса дискретизации: количество корректных дискретизаций и ошибок;
- Измерением среднеквадратического уровня собственного шума платы и построение гистограмм распределения значений сигнала и уровня шума. Статистическая обработка оцифрованного сигнала, как по банкам, так и для всего массива в целом;
- Управление цифро-аналоговыми преобразователями и формирование тестовых сигналов;
- Управление работой АЦП (включение нормального режима работы, или режима паттернов *);
- Работа с несколькими устройствами подключенных к ПК (до 4-х включительно). При этом обмен данными выполняется только с тем устройством, которое выбрано в программе (одновременный обмен данными с несколькими устройствами в программе на данный момент не предусмотрен);
- Сохранение сигналов в файле формата *.txt (текстовый файл);
- Сохранения и загрузка конфигурационного файла с установками программы, возможность установки настроек конфигурации по умолчанию;
- Работа в режиме модели (при отключенной плате).

* режим паттернов подразумевает переход АЦП в фиксированное состоянии в соответствии с технической документацией на АЦП.

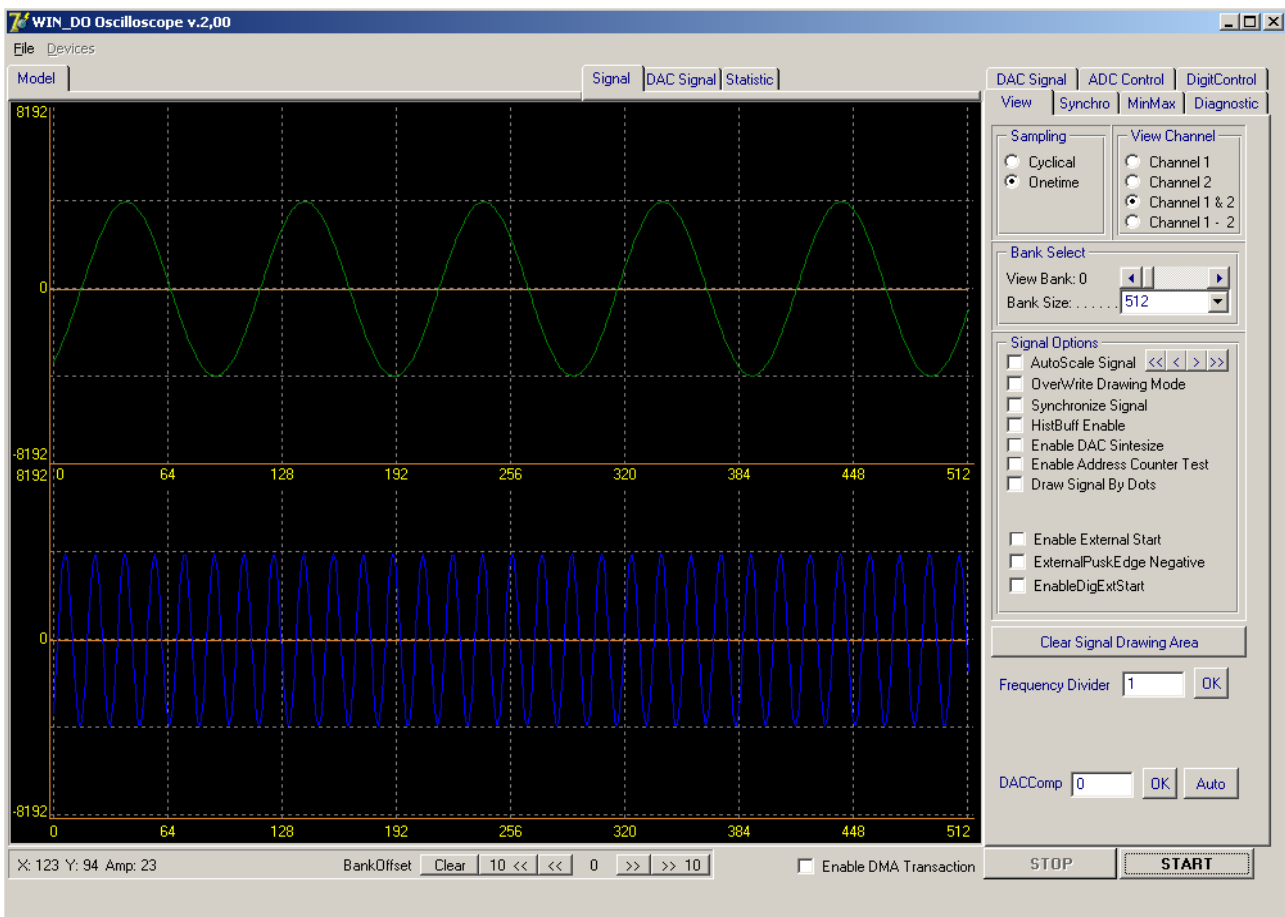


Рис. 1. Внешний вид программы «WIN_DO_125AS5_Oscilloscope»

Программа запускается выполнением файла "Win_do.exe" из каталога "WIN_DO_125AS5_Oscilloscope". Если устройство не обнаружено, то активной становится закладка «Model», и программа переходит в режим моделирования входного сигнала. При

обнаружении в системе подключенного устройства, на вкладке выбора текущего устройства появляется закладка «*Device*». Число этих закладок будет соответствовать числу подключенных устройств, но не более 4-ех, и каждая закладка отображает состояние сигналов непосредственно по каждому из устройств. Программа дает возможность настраивать работу устройства, подключенного к ПК.

Внимание: настройки, прописанные на одну плату автоматически применяются ко всем подключенным к ПК устройствам.

В нижнем правом углу окна программы находятся кнопки "Start" и "Stop" обеспечивающие пуск и останов процесса дискретизации, соответственно (рис. 1).

Левая часть окна программы содержит панель графического отображения сигналов и закладки для выбора текущего источника сигнала и режима работы программы. В правой части окна программы находится панель с вкладками, которые обеспечивают переключение между элементами управления программы сгруппированными по функциональному признаку.

Для смещения сигнала внутри отображаемого банка предназначена панель кнопок "BankOffset" (нижняя часть окна программы). Кнопка "Clear" позволяет сбросить смещение в ноль. Кнопки "<<10", "<<1", ">>1" и ">>10" позволяют задавать смещение в отсчетах для просмотра от начала текущего отображаемого банка. Данная панель позволяет более точно задать отображаемую область сигнала и работает во всех режимах работы с графиками.

«Enable DMA Transaction» - вкл\выкл. режим транзакций.

Закладка "View"

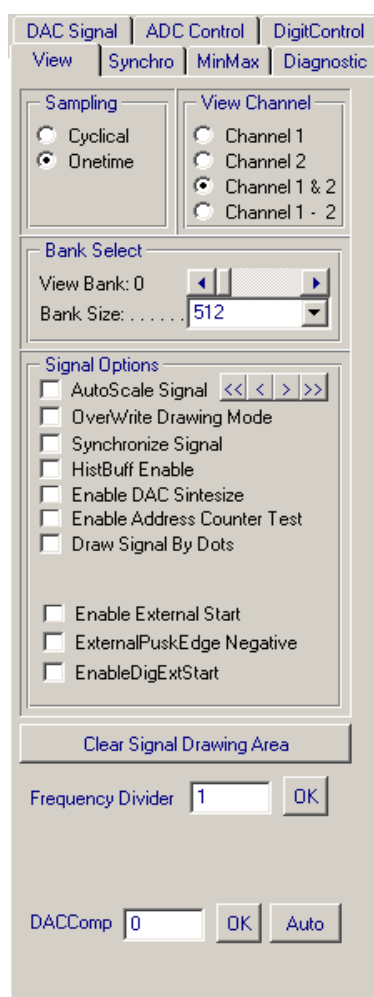


Рис. 2. Внешний вид закладки «View»

Закладка "View" содержит элементы для управления режимами отображения оцифрованного сигнала, а также элементы для управления платой.

Таблица 1. Элементы закладки "View"

Группа "Sampling"	Задаёт режим циклической (Cyclical) или однократной (Onetime) дискретизации.
Группа "View Channel"	Выбирает отображаемые каналы. - Channel 1 – просмотр 1-го канала. - Channel 2 – просмотр 2-го канала. - Channel 1 & 2 - одновременный просмотр 1(низ) и 2(верх) каналов. - Channel 1 - 2 – результат разницы между 1-ым и 2-ым каналами.
Группа "Bank Select"	Задаёт размер отображаемого банка и текущий банк. - View Bank - индицирует текущий отображаемый банк. Полоса прокрутки позволяет изменять текущий отображаемый банк для перемещения по оцифрованному сигналу во времени; - Bank Size - позволяет выбрать из списка или задать произвольно текущий размер банка (размер банка должен быть не менее 5).
Группа "Signal Options"	Позволяет выбрать следующие режимы: - AutoScale Signal - вкл.\выкл. режим автомасштабирования отображаемого сигнала. - Кнопки "<<", "<", ">" и ">>" служат для изменения коэффициента масштабирования вручную. При нажатии этих кнопок текущий коэффициент масштабирования будет отображаться в строке статуса. Кнопки действуют при выключенном режиме автоматического масштабирования « <i>Auto Scale Signal</i> » - OverWrite Drawing Mode - вкл.\выкл. режим наложения отображаемого сигнала (отображение сигнала без удаления предыдущего изображения); - Synchronize Signal - вкл.\выкл. режим программной синхронизации сигнала (параметры синхронизации задаются на вкладке <i>Synchro</i>). - HistBuff Enable - вкл.\выкл. на плате буфера предистории (используется при необходимости получения данных предшествовавших сигналу внешнего запуска). - Enable DAC Sintesize – вкл.\выкл. генерации сигнала ЦАП; - Enable Address Counter Test - вкл.\выкл. режим аппаратной подмены данных АЦП счетчиком (используется для диагностики работы внутренних схем платы); - Draw Signal By Dots - вкл.\выкл. прорисовку графика сигнала точками. - Enable External Start - вкл.\выкл. режим внешнего запуска процесса дискретизации; - ExternalPuskEdgeNegative - вкл.\выкл. внешнего запуска по заднему фронту. * - EnableDigExtStart – вкл.\выкл. внешнего запуска от цифрового входа,**
Кнопка "Clear Signal Drawing Area"	Позволяет отчистить область отображения сигнала, удобна для использования при включенном режиме наложения " <i>OverWrite Drawing Mode</i> ".
Frequency Divider	Задаёт значение делителя частоты дискретизации, устанавливается по

	нажатие кнопки "OK".
DACComp	Задаёт значение записываемое в ЦАП по нажатию кнопки «OK».
Кнопка «Auto»	Включает режим передачи в ЦАП значений адресного счетчика. ***

* режим действует как для сигнала запуска по аналоговому входу, так и по цифровому. Зависит от настроек сделанных пользователем в *EnableDigExtStart*.

** если в *Enable External Start* выключен режим внешнего запуска, то настройки, сделанные в *EnableDigExtStart* и *ExternalPuskEdgeNegative* учитываться программой не будут.

*** записываются значения от 0 до 4095 (обусловлено характеристиками ЦАП). Режим используется для тестирования устройства.

Закладка "Synchro"

Закладка предназначена для настройки параметров программной синхронизации относительно принимаемого сигнала. Операция выполняется выставлением порога синхронизации и указанием фронта, по которому будет выполняться программная синхронизация.

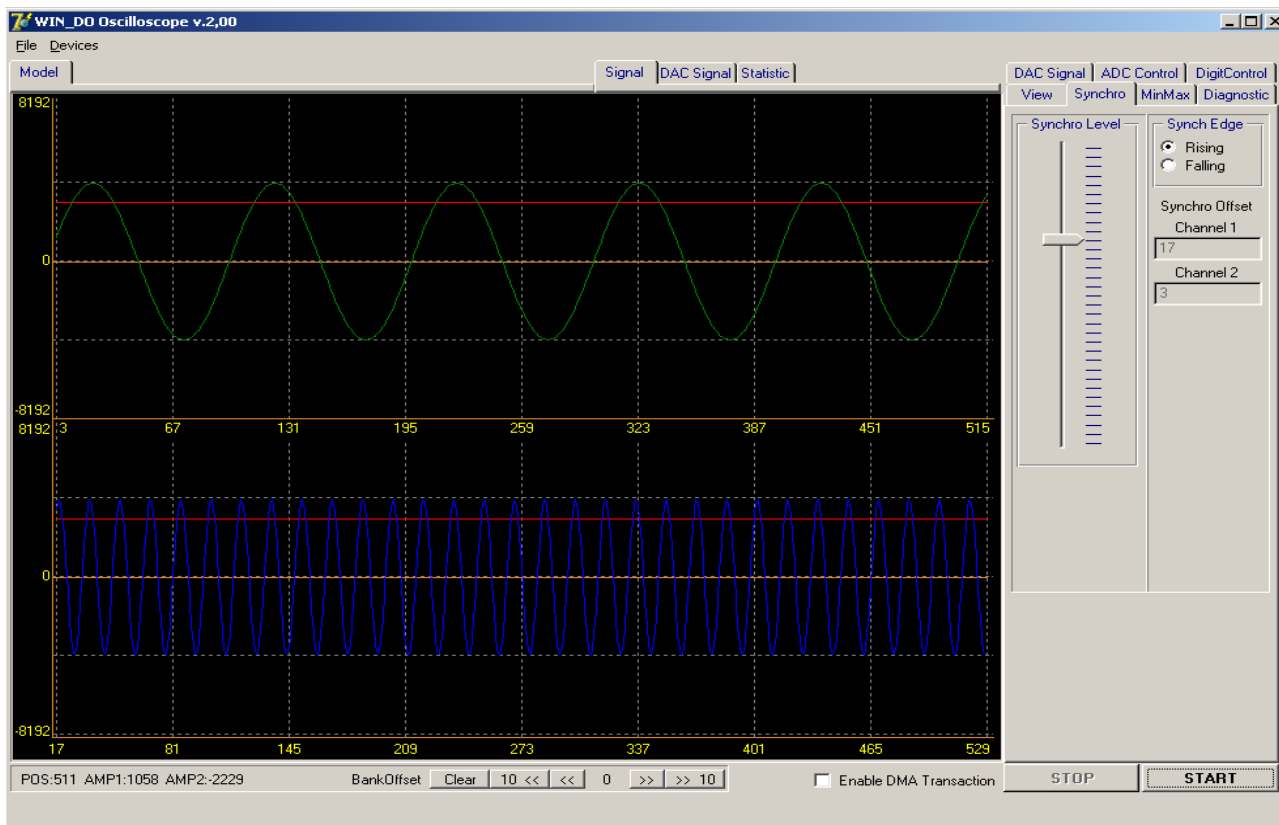


Рис. 3. Внешний вид окна программы с закладкой «Synchro»

Таблица 2. Элементы закладки «Synchro»

Группа "Synchro Level"	Позволяет с помощью ползунка задать порог синхронизации. Текущий порог отображается на сигнале горизонтальной красной линией*.
Группа "Synch Edge"	Позволяет выбрать фронт по которому будет выполняться программная синхронизация: - Rising - по возрастающему фронту; - Falling – по спадающему фронту.
«Synchro Offset»	Указывает значение задержки по временной оси при включенном режиме синхронизации (элемент «Synchronize Signal» на закладке «View»).

***Внимание:** при вкл. режиме «AutoScale Signal» графическое отображение уровня синхронизации не соответствует действительному.

Закладка "MinMax"

Информационная закладка отображающая минимальные и максимальные значения сигналов по каналам. Отображения идет как в активном банке (том, который выведен в данный момент на экран), так и по всему массиву сигнала.

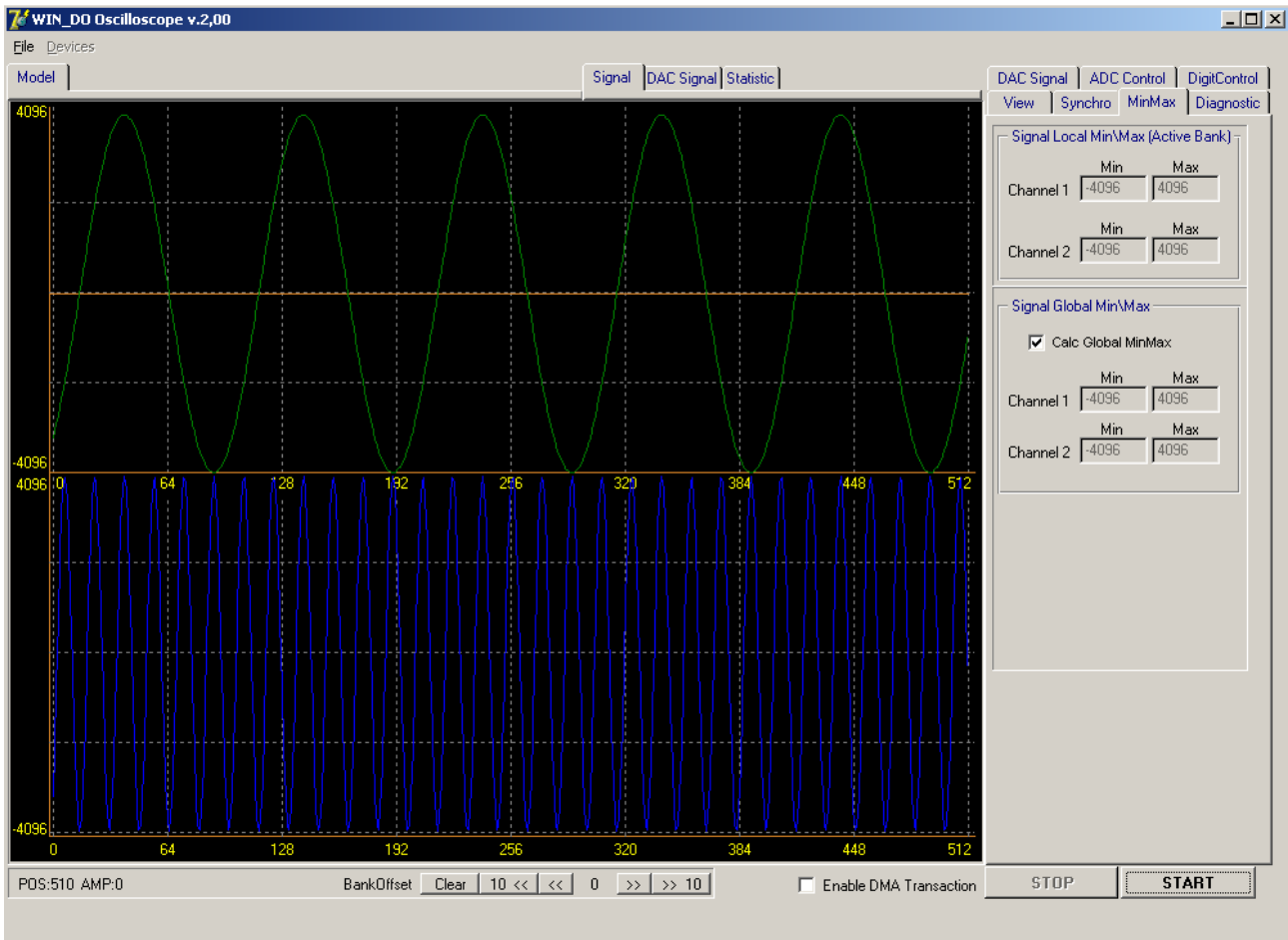


Рис. 4. Внешний вид закладки «MinMax»

Таблица 3. Элементы закладки "MinMax"

Группа "Signal Local MinMax"	Отображает текущие локальные (в пределах текущего отображаемого банка) минимум и максимум отдельно по каналам.
Группа "Signal Global MinMax"	Отображает текущие глобальные (по всему массиву оцифрованного сигнала) минимум и максимум отдельно по каналам. *

* Значения глобальных максимумов и минимумов рассчитываются при включенном режиме "CalcGlobalMinMax". По умолчанию режим отключен, т.к. он замедляет работу программы.

Закладка "Statistic"

Информационная закладка, где отображаются результаты статистической обработки оцифрованных данных.

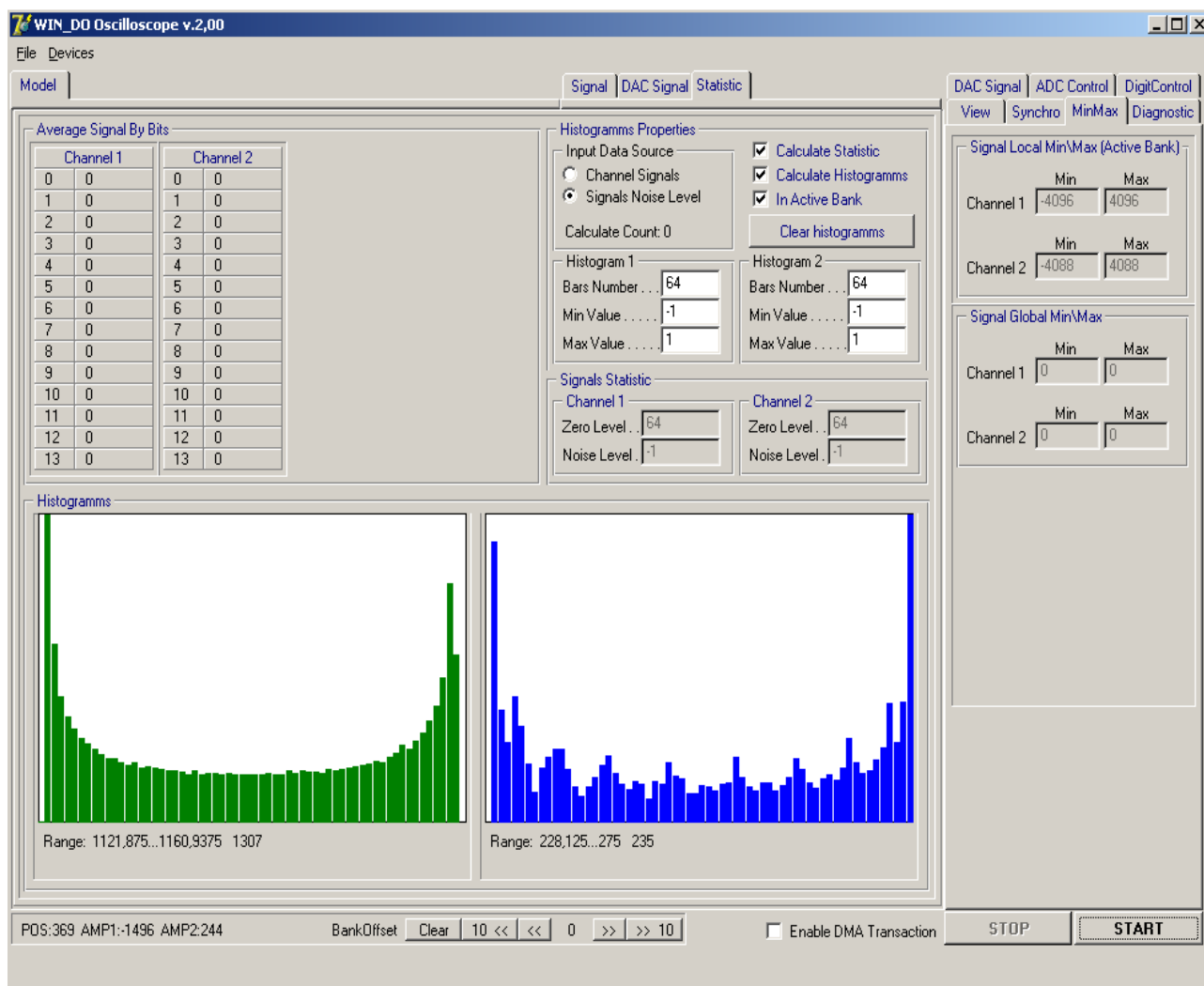


Рис. 5. Внешний вид окна программы с активной закладкой «Statistic»

Таблица 4. Элементы закладки «Statistic»

<p>Группа "Histogramms Properties"</p>	<p>Позволяет активизировать процесс вычисления статистических значений и гистограмм, а также задать параметры для их построения:</p> <ul style="list-style-type: none"> - Calculate Statistic - Вкл.\выкл. расчет уровня нуля и уровня шума по каналам; - Calculate Histogramms - Вкл.\выкл. построение гистограмм с текущими установленными параметрами (для работы этого режима необходимо чтобы был включен режим расчета статистических данных (режим <i>Calculate Statistic</i>)); - In Active Bank - Вкл.\выкл. обработка данных из активного банка; - Clear histogramms – (кнопка) позволяет очистить гистограммы и инициировать процесс накопление заново с новыми параметрами.
--	---

Группа "Input Data Source"	<p>Позволяет задать входные данные для построения гистограмм:</p> <ul style="list-style-type: none"> - Channel Signals - построение гистограмм будет осуществляться на основе оцифрованных данных по всей выборке (поканально); - Signals Noise Level - построение гистограмм будет осуществляться на основе рассчитанного уровня шума по каналам; - Calculate Count - отображает текущее число обработанных входных данных для построения гистограмм.
Группа "Histogram 1 (2)"	<p>Позволяет задать параметры для построения гистограмм:</p> <ul style="list-style-type: none"> - Bars Number - количество полосок разбиения диапазона значений гистограммы; - Min Value - минимальное значение диапазона значений гистограммы; - Max Value - максимальное значение диапазона значений гистограммы.
Группа "Signal Statistic"	<p>Отображает рассчитанный средний уровень и уровень шума сигналов по каналам (для работы этого режима необходимо чтобы был включен режим расчета статистических данных - режим <i>Calculate Statistic</i>).</p>

Закладка «DAC Signal»

Закладка предназначена для тестирования ЦАП-ов, установленных на плате. Для этого на закладки имеется возможность выбора типа тестирующего сигнала, подаваемого на ЦАП, а также можно настроить временные и амплитудные параметры этих сигналов.

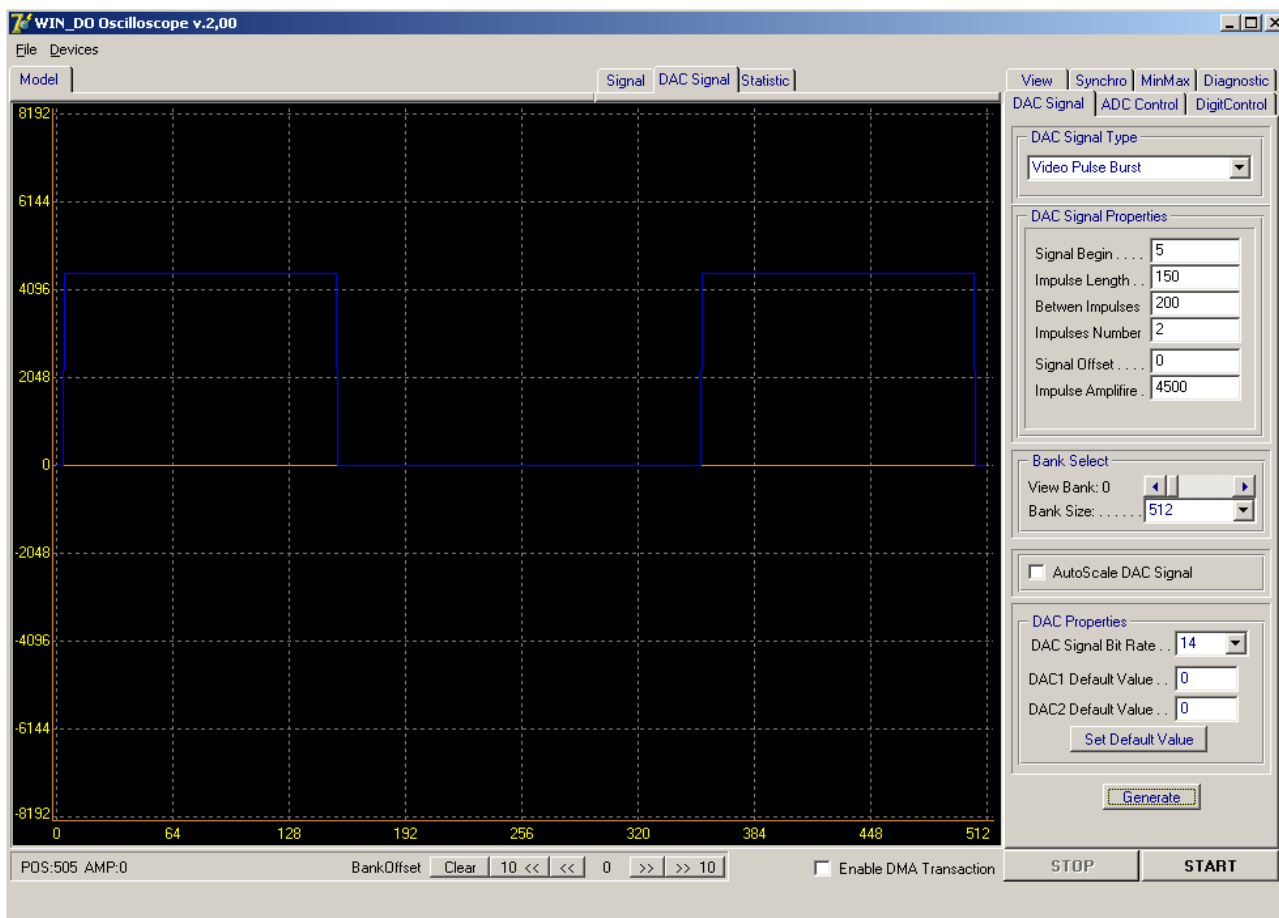


Рис. 6. Внешний вид программы с открытой вкладкой «DAC Signal»

Таблица 5. Элементы закладки «DAC Signal»

«DAC Signal Type»	Позволяет выбрать тип сигнала генерируемого программой и подаваемого на ЦАП (Video Pulse, Video Pulse Burst, Radio Pulse, Radio Pulse Burst, LCHM Pulse, Barcer Code).
Группа «DAC Signal Properties»	Позволяет указать параметры сигнала, передаваемого на ЦАП: - Signal Begin – задает начальную точку, с которой формируется сигнал; - Impulse Length – задает длину импульса в отсчетах; - Betwen Impulse – задает интервал между импульсами в отсчетах (расстояние между задним фронтом предыдущего импульса и передним фронтом последующего импульса); - Impulse Number – задает количество генерируемых импульсов; - Signal Offset – задает значение смещения нулевого уровня генерируемого сигнала; - Impulse Amplifize – задает значение амплитуды импульса

	(максимальное допустимое значение 8192).
Группа «Bank Select»	<p>Задаёт размер отображаемого банка и текущий банк.</p> <p>- View Bank - индицирует текущий отображаемый банк. Полоса прокрутки позволяет изменять текущий отображаемый банк для перемещения по оцифрованному сигналу во времени;</p> <p>- Bank Size - позволяет выбрать из списка или задать произвольно текущий размер банка (размер банка должен быть не менее 5).</p>
«Auto Scale DAC Signal»	Вкл.\Выкл. режима автомасштабирования отображаемого сигнала на ЦАПе.
Группа «DAC properties»	<p>Позволяет настроить параметры сигнала, устанавливаемые по умолчанию:</p> <p>- DAC Signal Bit Rate - задаёт размерность цифрового сигнала на входе ЦАП (в программе доступен один вариант – 14 бит);</p> <p>- DAC1 Default Value - задаёт значение сигнала на входе ЦАП1 (добавляется к смещению нулевого уровня сигнала указанному в <i>Signal Offset</i>);</p> <p>- DAC2 Default Value - задаёт значение сигнала на входе ЦАП2 (добавляется к смещению нулевого уровня сигнала указанному в <i>Signal Offset</i>);</p> <p>- Set Default Value - (кнопка) позволяет записать установленные параметры, как используемые по умолчанию.</p>
Кнопка «Generate»	Позволяет запустить процесс генерации сигналов для ЦАП. После генерации программа запросит разрешение на передачу сгенерированных сигналов для ЦАП1 и ЦАП2. *

* При записи сигнала ЦАП установленного устройства предусмотрена возможность раздельной записи. Т.е. сгенерированный сигнал можно записать в один из ЦАП-ов по выбору, либо в оба.

Закладка «Diagnostic»

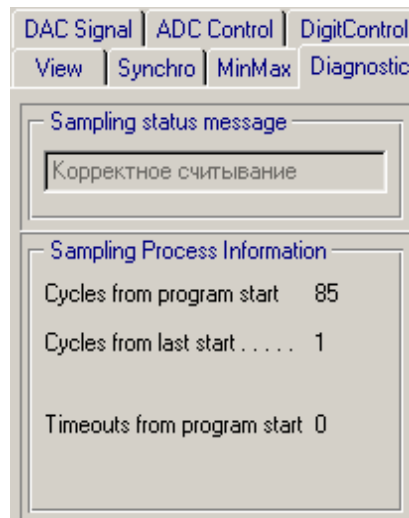


Рис. 7. Внешний вид закладки «Diagnostic» при включенной плате

Таблица 6. Элементы закладки «Diagnostic»

Группа "Sampling Status Message"	В случае корректного завершения текущего процесса дискретизации отображается строка " <i>Корректное считывание</i> ", в противном случае — " <i>Нет готовности</i> ";
Группа "Sampling Process Information"	Отображает информацию о количестве успешных и ошибочных дискретизаций; - Cycles from program start - число корректных считываний от старта программы; - Cycle from last start - число корректных считываний от последнего нажатия кнопки " <i>Start</i> "; - Timeouts from program start - число некорректных считываний от старта программы (число тайм-аутов ожидания прерывания);

Закладка «ADC Control»

На закладке имеются инструменты для тестирования работы АЦП, установленных на плате. При этом можно устанавливать как нормальный режим работы, так и работа по паттернам, когда АЦП начинает генерировать сигнал заданной формы.

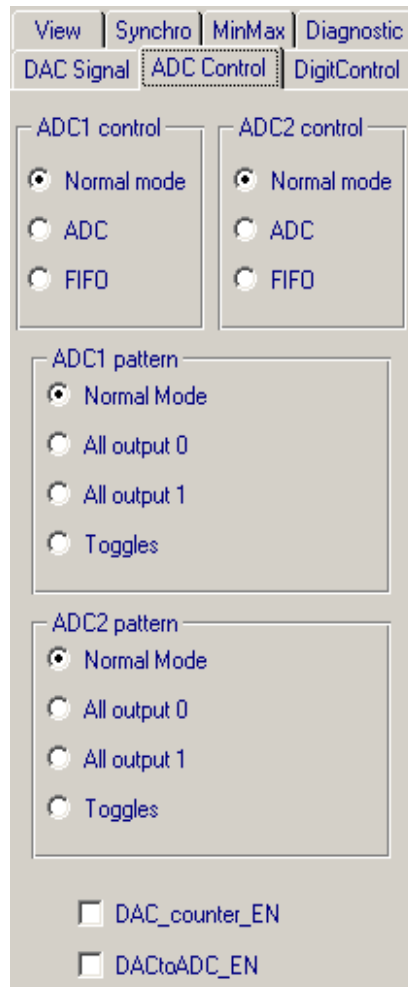


Рис. 8. Внешний вид закладки «ADC Control»

Таблица 7. Элементы закладки «ADC Control»

Группа «ADC1 Control»	Предназначен для выбора режима тестирования АЦП1 - Normal Mode - установка нормального режима работы; - ADC - установка режима, когда вместо данных АЦП имитируется адресный счетчик; - FIFO - установка режима, когда вместо внутреннего FIFO, данные передаются с адресного счетчика.
Группа «ADC2 Control»	Предназначен для выбора режима тестирования АЦП2 - Normal Mode - установка нормального режима работы; - ADC - установка режима, когда вместо данных АЦП имитируется адресный счетчик; - FIFO - установка режима, когда вместо внутреннего FIFO, данные передаются с адресного счетчика.
Группа «ADC1 pattern»	Предназначена для настройки АЦП1 на режим передачи по паттернам:

	<ul style="list-style-type: none"> - Normal Mode - установка нормального режима работы при котором выполняется дискретизация данных, поступающих на вход АЦП1; - All output 0 - установка режима паттернов, при котором все цифровые выходы АЦП1 устанавливаются в «0»; - All output 1 - установка режима паттернов, при котором все цифровые выходы АЦП1 устанавливаются в «1»; - Toggles - установка режима паттернов, при котором на каждом такте происходит инверсия цифровых выводов АЦП1.
Группа «ADC2 pattern»	<p>Предназначена для настройки АЦП2 на режим передачи по паттернам:</p> <ul style="list-style-type: none"> - Normal Mode - установка нормального режима работы при котором выполняется дискретизация данных, поступающих на вход АЦП2; - All output 0 - установка режима паттернов, при котором все цифровые выходы АЦП2 устанавливаются в «0»; - All output 1 - установка режима паттернов, при котором все цифровые выходы АЦП2 устанавливаются в «1»; - Toggles - установка режима паттернов, при котором на каждом такте происходит инверсия цифровых выводов АЦП2.
«DAC_counter_EN»	вкл.\выкл. режима передачи счетчика на ЦАП. Используется в тестовом режиме.
«DACtoADC_EN»	вкл.\выкл. режима непосредственной передачи данных с памяти ЦАП в память АЦП устройства. Используется в тестовом режиме.

Закладка «DigitControl»

Закладка предназначена для отображения цифровых сигналов платы, а также для управления значением этих цифровых сигналов.

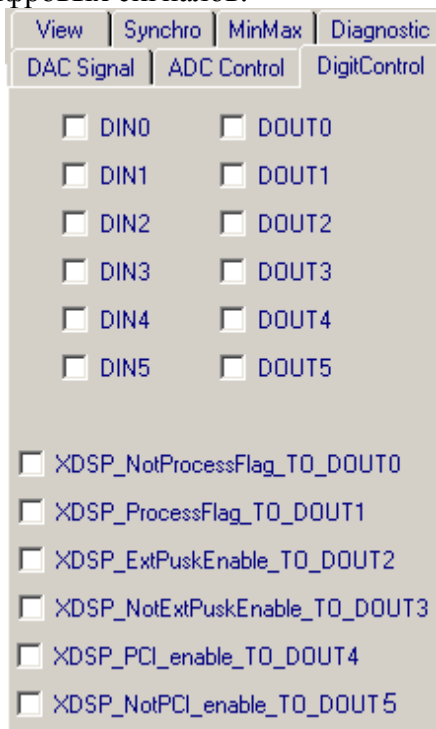


Рис. 9. Внешний вид закладки «DigitControl»

Таблица 8. Элементы закладки «DigitControl»

Группа «DIN»	Отображает состояние цифровых сигналов на плате. Выставляется автоматически. «DIN5» сигнал цифрового запуска; остальные не используются.
Группа «DOUT»	Позволяет пользователю вручную настраивать цифровые выходы платы.
«XDSP_NotProcessFlag_TO_DOUT0»	Передает на цифровой вывод «DOUT0» инвертированное значение флага оцифровки данных (временное окно).
«XDSP_ProcessFlag_TO_DOUT1»	Передает на цифровой вывод «DOUT1» значение флага оцифровки данных (временное окно).
«XDSP_ExtPuskEnable_TO_DOUT2»	Передает на цифровой вывод «DOUT2» значение сигнала внешнего запуска оцифровки данных.
«XDSP_NotExtPuskEnable_TO_DOUT3»	Передает на цифровой вывод «DOUT3» инвертированное значение сигнала внешнего запуска оцифровки данных.
«XDSP_PCI_enable_TO_DOUT4»	Передает на цифровой вывод «DOUT4» сигнал DMA-транзакции.*
«XDSP_NotPCI_enable_TO_DOUT5»	Передает на цифровой вывод «DOUT5» инвертированное значение сигнала DMA-транзакции.

* при включенном режиме внешнего запуска сигнал находится в состоянии лог. «1» до поступления сигнала внешнего запуска, после чего, переходит в лог. «0». Начало оцифровки данных на АЦП сигнализирует сигнал «временное окно», который устанавливается в состояние лог. «1».

Меню "File"

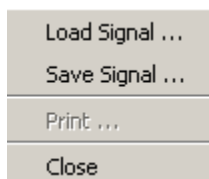


Рис. 10. Меню «File»

1). Сохранение и считывание данных.

Для хранения данных в программе используется текстовый файл. Ограничений на имя файла и его расширение не установлено. Программа позволяет сохранять и считывать данные 2-х каналов, как вместе, так и по отдельности в зависимости от настроек на закладке «View», группа «View Channel». Все данные записываются в один файл.

Для чтения данных используется меню «File/Load Signal...»; для записи сигнала в файл используется меню «File/Save Signal...».

2). Закрытие программы

Меню “Close” закрывает приложение.

Меню “Devices”



Рис. 11. Меню “Devices”

Предназначено для выбора устройства, подключенного к ПК. Программа позволяет работать с четырьмя подключенными платами. В меню напротив выбранного устройства ставится соответствующая метка, а в области графиков отображаются соответствующие сигналы.

Описание библиотеки "Xdspapi.dll"

Библиотека "xdspapi.dll" предназначена для использования платы в прикладных программах. Библиотека реализует высокоуровневые API функции для управления режимами работы платы и получения оцифрованных данных из буферной памяти. Вместе с библиотекой поставляются: lib-файл *xdspapi.lib* для использования библиотеки под С-компилятором (например, Visual C++), заголовочные файлы *xdspapi.pas* и *xdspapi.h* для подключения библиотеки к проекту на Pascal и С соответственно.

В комплекте программного обеспечения поставляемого с платой (каталог WIN_DO_125AS5_Oscilloscope) находится проект в исходных текстах демонстрирующий работу с библиотекой в Delphi.

Плата поддерживает работу PCI-express интерфейса в режиме "Master/Slave". В этом режиме плата производит копирование внутренней памяти в физическую память компьютера и из памяти компьютера во внутреннюю память устройства без участия центрального процессора, так называемый DMA (*Direct Memory Access*) режим. Работа платы в режиме "Master/Slave" обеспечивает обмен данными между внутренней памятью платы и памятью компьютера на скорости до 100 MB/Sec.

При работе платы в режиме "Master/Slave" используется следующий механизм пересылки оцифрованного сигнала из памяти АЦП в память компьютера:

- при инициализации платы, в оперативной памяти компьютера выделяется 2 буфера (DMA буфер) по 32 МБ каждый (для хранения данных 2-х каналов);
- после завершения процесса дискретизации плата автоматически производит копирование участка внутренней памяти АЦП (длина копируемого участка задается) в DMA 1-й буфер компьютера, без участия центрального процессора;
- оставшийся DMA буфер копируется непосредственно по команде старта DMA транзакции;
- пользовательское приложение получает доступ к буферу напрямую, через указатель на DMA буфер.

При пересылке сигнала из памяти компьютера в память ЦАП используется следующий механизм:

- при инициализации платы, в оперативной памяти компьютера выделяется 2 буфера (DMA буфер) по 32 МБ каждый (для хранения данных 2-х каналов);
- после запуска процесса передачи данных данные из DMA 3-го и 4-го буфера пересылаются в память ЦАП на плате устройства по команде старта DMA транзакции.
- пользователь получает доступ к буферу напрямую через указатель DMA буфера.

Плата поддерживает генерацию аппаратного прерывания. Прерывание генерируется после завершения процесса DMA транзакции.

Основной режим работы платы - цифровой осциллограф с циклическим запуском процесса оцифровки.

При запуске в памяти компьютера выделяется 4-е DMA буфера для обмена данными с платой. Два из этих буферов идут на каналы АЦП, и два на каналы ЦАП. Оцифрованные данные (2 канала АЦП) сохраняются в буферной памяти платы и, затем, по команде *XdspStartRead* передаются в один из 2-х DMA буферов компьютера для обработки. 3-й и 4-й DMA буферы компьютера используются для передачи данных на ЦАП (2 канала ЦАП). Процесс передачи осуществляется командой *XdspStartWrite*.

DLL библиотека "xdspapi.dll" подключается к проекту программы стандартными методами для используемой среды программирования.

Модуль "xdspapi.pas" (*xdspapi.h*), описывающий импортируемые функции, а также дополнительные константы и типы для среды программирования Delphi (Visual C++), поставляется в комплекте программного обеспечения с платой.

Библиотека поддерживает следующие группы функций:

- открытие (инициализация), закрытие (деинициализация) платы;
- запуск процесса дискретизации и DMA транзакции;
- управление режимом внешнего запуска;
- настройка процесса оцифровки данных;
- обмен данными с АЦП и ЦАП;
- обработка цифровых сигналов.

Перечень функций:

- открытие (инициализация), закрытие (деинициализация) платы:
 - XdspOpenDevice** — Открыть устройство
 - XdspCloseDevice** - Закрыть устройство
 - XdspResetAll** - Выполнить RESET платы
- запуск процесса дискретизации и DMA транзакции:
 - XdspStartWrite** — запись данных в ЦАП
 - XdspStartRead** — чтение данных из АЦП
- управление режимом внешнего запуска:
 - XdspExternalStartEnable** - включение режима внешнего запуска
 - XdspExternalStartDisable** - выключение режима внешнего запуска
 - XdspSetPosEdgeExtStart** – включение внешнего запуска по положительному фронту
 - XdspSetNegEdgeExtStart** – включение внешнего запуска по отрицательному фронту
 - XDSPDigExtStartEnable** — включение режима внешнего запуска по цифровому сигналу
 - XDSPDigExtStartDisable** — выключение режима внешнего запуска по цифровому сигналу
- настройка процесса оцифровки данных.
 - XdspSetProcessLength** - установить количество отсчетов дискретизации и длину DMA транзакции
 - XdspSetSampleFrequencyDivider** - установить делитель частоты процесса оцифровки
 - XdspAddrCntrToADCDDataEnable** - включение режима подмены данных АЦП адресным счетчиком
 - XdspAddrCntrToADCDDataDisable** - выключение режима подмены данных АЦП адресным счетчиком
 - XdspSetDAC1Default** – запись данных в ЦАП1 установленных по умолчанию
 - XdspSetDAC2Default** – запись данных в ЦАП2 установленных по умолчанию
 - XdspHistBuffEnable** – включение буфера истории
 - XdspHistBuffDisable** – выключение буфера истории
- обмен данными с АЦП и ЦАП.
 - XdspDACCounterEnable** – подключение счетчика к ЦАП-у
 - XdspDACCounterDisable** – отключение счетчика от ЦАП-а
 - XdspDACtoADCEnable** – включает непосредственный обмен данными между памятью ЦАП и АЦП на плате
 - XdspDACtoADCDisable** – выключает непосредственный обмен данными между памятью ЦАП и АЦП на плате
 - XdspWriteToADC DAC** – запись слова в конфигурационные регистры АЦП и ЦАП
 - XdspDACEnable** - включение генерации сигнала ЦАП
 - XdspDACDisable** - выключение генерации сигнала ЦАП
- обработка цифровых сигналов.
 - XDSPReadDigInData** - чтение цифровых сигналов с платы
 - XDSPWriteDigOutData** - запись цифровых сигналов в плату.

Function XdspOpenDevice - Открыть устройство

Декларация: **Function XdspOpenDevice(bDeviceNumber:Byte;
phInterruptEventHandle:PHandle; ppDmaBuffer1:PPDmaMemoryBuffer;
ppDmaBuffer2:PPDmaMemoryBuffer; ppDmaBuffer3:PPDmaMemoryBuffer;
ppDmaBuffer4:PPDmaMemoryBuffer; pDmaBufferAddr1:PDWord;
pDmaBufferAddr2:PDWord; pDmaBufferAddr3:PDWord;
pDmaBufferAddr4:PDWord) : TDeviceHandle**

Параметры функции: bDeviceNumber - номер устройства;
phInterruptEventHandle - указатель на прерывание от устройства;
ppDmaBuffer1 — указатель на 1-й DMA буфер компьютера;
ppDmaBuffer2 — указатель на 2-й DMA буфер компьютера;
ppDmaBuffer3 — указатель на 3-й DMA буфер компьютера;
ppDmaBuffer4 — указатель на 4-й DMA буфер компьютера;
pDmaBufferAddr1 — указатель на физический адрес 1-го DMA буфера;
pDmaBufferAddr2 — указатель на физический адрес 2-го DMA буфера;
pDmaBufferAddr3 — указатель на физический адрес 3-го DMA буфера;
pDmaBufferAddr4 — указатель на физический адрес 4-го DMA буфера.

Функция предназначена для открытия и инициализации выбранной платы. Возвращает дескриптор открытой платы, который передается в остальные функции библиотеки как определитель устройства к которому обращается программа. Функция получает указатель на DMA буфер компьютера и на физический адрес DMA, получает дескриптор для обработки событий по прерыванию от платы, разрешает прерывания после завершения DMA транзакций и включает автоматическое выполнение операций по DMA транзакции. Выполняет очистку регистров.

В случае ошибки возвращается значение `XDSP_INVALID_DEVICE_HANDLE` (см. константы и типы для работы с API функциями библиотеки, описанные в файле "xdspapi.pas"), код ошибки можно получить используя стандартную Windows API функцию `GetLastError`.

Примечание: при инициализации платы значение делителя частоты дискретизации устанавливается равным 1 (см. функцию *XdspSetSampleFrequencyDivider*), размер процесса дискретизации и DMA транзакции (см. функцию *XdspSetProcessLength*) устанавливается равным значению `dwLength`. Эти параметры могут быть изменены в процессе работы программы вызовом функций *XdspSetSampleFrequencyDivider* и *XdspSetProcessLength*.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменные для хранения дескриптора устройства
  device:TDeviceHandle;
  hdv:THandle;
  // Указатель на номер подключенной платы
  i : integer;
  ...

  // Активация устройств подключенных к ПК (по умолчанию принято, что число
  устройств может быть равным до 4 штук)
  if i:=1 to 4 do begin
```

```
// Открыть плату (нумерация плат 0-3)
hdv:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
@XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
@XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
@XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
@XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

// Если устройство открыто ошибочно
If device=XDSP_INVALID_DEVICE_HANDLE Then Begin
    // Код ошибки может быть получен функцией GetLastError.

    exit;
end;

// формирование указателя открытой платы
device(i) := hdv;

// Устройство открыто и проинициализировано успешно
end;

...

// Работа с платой
```

Function XdspCloseDevice - Закрыть устройство

Декларация: **Function XdspCloseDevice(hDeviceHandle:TDeviceHandle):DWORD**

Функция предназначена для закрытия платы ранее открытой функцией XdspDeviceOpen. Закрытие платы выполняется перед завершением работы программы пользователя.

Пример использования функции в среде Delphi :

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменные для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель на номер подключенной платы
  i : ineteger;

...

  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
  функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
  device(i) := hdv;

...

  // Работа с платой

...

  // Выбор устройства для закрытия (возможно использование до 4-ех устройств)
  if i:=1 to 4 do begin
    // Закрываем плату перед завершением работы
    XdspCloseDevice(device(i));
  end;

End;
```

Function XdspResetAll - Выполнить RESET платы

Декларация: **Function XdspResetAll(hDeviceHandle:TDeviceHandle):DWORD**

Функция предназначена для внутренней инициализации платы.

Примечание: для устойчивой и надежной работы платы, рекомендуется, посредством выполнения функции *XdspResetAll*, проводить инициализацию (реинициализацию) платы, каждый раз перед вызовом функций *XdspStartWrite* или *XdspStartRead*.

Параметры функции: hDeviceHandle - дескриптор открытой платы.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель на номер подключенной платы
  i : integer;

...

  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

...

  // Работа с платой
  // Инициализирует плату
  XdspResetAll(device);

...

  // Закрываем плату перед завершением работы, выбор номера устройства опущена
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```


Function XdspStartWrite – запускает процесс передачи данных из памяти компьютера в буферную память платы (ЦАП)

Декларация: **Function XDspStartWrite(hDeviceHandle:TDeviceHandle; BufNumber:integer; RamNumber:integer; TimeOut:DWord):DWord**

Функция разрешает чтение DMA транзакций перед пересылкой сгенерированных данных на ЦАП платы. Запускает DMA транзакцию, при которой диапазон внутренней памяти компьютера копируется во внутреннюю память платы (память ЦАП; Используются буферы под номерами 2 и 3). После завершения DMA транзакции функция запрещает чтение DMA транзакции.

Функция возвращает XDSP_STATUS_SUCCESS в случае успешного выполнения или XDSP_STATUS_ERROR в случае превышения таймаута (см. константы и типы для работы с API функциями библиотеки, описанные в файле "xdspapi.pas"), код ошибки можно получить используя стандартную Windows API функцию GetLastError.

Параметры функции: hDeviceHandle — дескриптор открытой платы;
BufNumber — номер буфера внутренней памяти платы (для ЦАП принимает значения 2 или 3);
RamNumber — номер DMA буфера (принимает значения 2 или 3);
TimeOut — время ожидания прихода прерывания от платы.

Пример использования функции в Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Переменная для хранения результата работы функции
  result : Dword;
  // указатель на номер подключенной платы
  i : integer;
...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
  функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
...
  // Работа с платой
  // Инициализирует плату, проверка ошибок опущена, см. функцию XdspResetAll
  XdspResetAll(device);
  // Запуск копирования данных со 2-го DMA буфера компьютера во 2-й буфер платы,
  // Время ожидания таймаута 20с
  result:=XdspStartWrite(device,2,2,20000);
  // Если ошибка
  If result=XDSP_STATUS_ERROR Then Begin
    // Произошла ошибка при выполнении функции, код ошибки можно получить
    используя GetLastError
  End;
  // Аналогично с 3-им буфером
...
  // Закрываем плату перед завершением работы, выбор номера устройства опущен
  (см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XdspStartRead – запускает процесс передачи данных из памяти платы (АЦП) в память компьютера

Декларация: **Function XDspStartRead(hDeviceHandle:TDeviceHandle; TimeOut:DWord; DMATrCB:DWord):DWord**

Функция, в зависимости от выбора пользователя в интерфейсе программы, запускает DMA транзакцию при которой диапазон внутренней памяти платы (АЦП) копируется во внутреннюю память компьютера (используются буферы с номерами 0 и 1), или функция запускает процесс дискретизации, по завершении которого автоматически производится запуск DMA транзакции. Функция ожидает прерывание, которое генерируется платой и сигнализирует о завершении DMA транзакции.

Функция возвращает XDSP_STATUS_SUCCESS в случае успешного выполнения или XDSP_STATUS_ERROR в случае превышения времени таймаута(см. константы и типы для работы с API функциями библиотеки, описанные в файле "xdspapi.pas"), код ошибки можно получить используя стандартную Windows API функцию GetLastError.

Примечание: после процесса дискретизации автоматически производится запуск DMA транзакции. Если время ожидания прихода прерывания от платы превышено, то выводится сообщение об ошибке.

Параметры функции: hDeviceHandle — дескриптор открытой платы;
TimeOut — время ожидания прихода прерывания от платы;
DMATrCB — указатель на выполняемую операцию (запуск DMA транзакции или запуска процесса дискретизации).

Пример использования функции в Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Переменная для хранения результата работы функции
  result : Dword;
  // Указатель на номер подключенной платы
  i : integer;
  // Указатель операции
  DMATrCB : Dword;

  ...

  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
  функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

  ...

  // Работа с платой
  // Инициализирует плату, проверка ошибок опущена, см. функцию XdspResetAll
  XdspResetAll(device);
  // Выбор операции в зависимости от настроек в окне интерфейса. За выбор
  // отвечает компонент «DMATransactionCheckBox»
```

```
if DMATransactionCheckBox.Checked = true then DMATrCB:=1
else DMATrCB:=0; end;

// Запуск передачи данных из памяти платы в память компьютера
result:=XDspStartRead(device,3000,DMATrCB); // время ожидания прерывания 3 с

// Если ошибка
If result=XDSP_STATUS_ERROR Then Begin
    // Произошла ошибка при выполнении функции, код ошибки можно получить
    используя GetLastError
End;

...

// Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
XdspCloseDevice(device);
```

Function XdspExternalStartEnable - Включение режима внешнего запуска

Декларация: **Function XdspExternalStartEnable(hDeviceHandle:TDeviceHandle):DWORD**

Функция разрешает режим внешнего запуска платы. В режиме внешнего запуска, функция *XdspSampleStart* является разрешающей для процесса дискретизации. При этом сам процесс дискретизации запускается переходом сигнала, на входе внешнего запуска, из состояния логический «0» («1») в состояние логической «1» («0») (уровни *LVTTL*) в зависимости от установленного запускающего фронта. Дальнейшее поведение платы (режим автоматического запуска DMA транзакции, генерация прерывания) аналогично работе с программным стартом процесса дискретизации.

Примечание: При работе платы в режиме внешнего запуска тайм-аут ожидания события по прерыванию (переменная *IntrWaitTimeOut*) устанавливается равным "бесконечности". В этом случае при отсутствии сигнала внешнего запуска программа будет зависать. В случае необходимости в тексте программы можно установить максимальное время ожидания прихода сигнала внешнего запуска.

Параметры функции: hDeviceHandle - дескриптор открытой платы.

Пример использования функции в среде Delphi показан при описании функции *XdspExternalStartDisable* (см. следующую страницу).

Function XdspExternalStartDisable - Выключение режима внешнего запуска

Декларация: **Function XdspExternalStartDisable(hDeviceHandle:TDeviceHandle):DWORD**

Функция выключает режим внешнего запуска платы.

Параметры функции: hDeviceHandle - дескриптор открытой платы.

Пример использования функций *XdspExternalStartEnable* и *XdspExternalStartDisable* в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TDeviceHandle;
  // Переменная для хранения результатов работы функции
  result : Dword;
  // указатель на номер подключенной платы
  i : integer;
...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

...
  // Работа с платой
  // Включить режим внешнего запуска
  XdspExternalStartEnable(device);

...

  // Запустить процесс дискретизации с внешним запуском, бесконечный тайм-аут -
ожидание прихода сигнала внешнего запуска
  result:=XdspSampleStart(device, INFINITE);

...
  //Обработка полученных данных
...

  // Выключить режим внешнего запуска
  XdspExternalStartDisable(device);

...
  // Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XDSPSetPosEdgeExtStart – Включение режима внешнего запуска по положительному фронту

Декларация: **Function XDSPSetPosEdgeExtStart(hDeviceHandle:TDeviceHandle):DWORD**

Функция определяет по какому фронту сигнала внешнего запуска будет выполняться пуск платы.

Данная функция для выбранного устройства разрешает запуск по переднему фронту сигнала внешнего пуска.

Функция действует как для аналогового, так и для цифрового сигналов внешнего запуска.

Выбор типа сигнала внешнего запуска осуществляется при помощи функций *XDSPDigExtStartEnable* и *XDSPDigExtStartDisable*.

Параметры функции: hDeviceHandle – дескриптор открытой платы

Пример использования функции в среде Delphi показан при описании функции *XDSPSetNegEdgeExtStart* (см. следующую страницу).

Function XDSPSetNegEdgeExtStart – Включение режима внешнего запуска по отрицательному фронту

Декларация: **Function XDSPSetNegEdgeExtStart(hDeviceHandle:TDeviceHandle):DWORD**

Функция определяет по какому фронту сигнала внешнего запуска будет выполняться пуск платы. Для выбранного устройства разрешается запуск по заднему фронту сигнала внешнего запуска.

Функция действует как для аналогового, так и для цифрового сигналов внешнего запуска. Выбор типа сигнала внешнего запуска осуществляется при помощи функций *XDSPDigExtStartEnable* и *XDSPDigExtStartDisable*.

Параметры функции: hDeviceHandle – дескриптор открытой платы

Пример использования функций *XDSPSetPosEdgeExtStart* и *XDSPSetNegEdgeExtStart* в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TDeviceHandle;
  // Указатель режима внешнего запуска (положительный или отрицательный фронт)
  ExternalPuskEdge : boolean;
  // указатель на номер подключенной платы
  i : integer;
  ...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
  функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
  ...
  // Выбор режима запуска
  if ExternalPuskEdge = true then
    XDSPSetNegEdgeExtStart(device);
  else
    XDSPSetPosEdgeExtStart(device);
  end;
  // Работа с платой
  // Включить режим внешнего запуска
  XdspExternalStartEnable(device);
  ...
  // Запустить процесс дискретизации с внешним запуском, бесконечный тайм-аут -
  ожидание прихода сигнала внешнего запуска
  result:=XdspSampleStart(device, INFINITE);
  ...
  //Обработка полученных данных
  ...
  // Выключить режим внешнего запуска
  XdspExternalStartDisable(device);
  ...
  // Закрываем плату перед завершением работы, выбор устройства опущены (см.
  функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XDSPDigExtStartEnable – Включение режима внешнего запуска по цифровому сигналу

Декларация: **Function XDSPDigExtStartEnable(hDeviceHandle:TDeviceHandle):DWORD**

Функция предназначена для включения на выбранной плате режима внешнего запуска от цифровых входов.

Плата может работать в двух режимах внешнего запуска — от аналогового входа или от цифрового. Реакция на сигналы внешнего запуска может осуществляться по переднему или по заднему фронту, выбор которых описан в функциях *XDSPSetPosEdgeExtStart* и *XDSPSetNegEdgeExtStart*.

Параметры функции: hDeviceHandle — дескриптор открытой платы.

Пример использования функции в среде Delphi показан при описании функции *XDSPDigExtStartDisable* (см. следующую страницу).

Function XDSPDigExtStartDisable – Выключение режима внешнего запуска по цифровому сигналу

Декларация: **Function XDSPDigExtStartDisable(hDeviceHandle:TDeviceHandle):DWORD**

Функция выключает режим запуска оцифровки на выбранной плате по внешнему цифровому входу.

Параметры функции: hDeviceHandle — дескриптор открытой платы.

Пример использования функций *XDSPDigExtStartEnable* и *XDSPDigExtStartDisable* в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Const
  TransactionLength = 262144; //1 М отсчетов на канал

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // указатель на номер подключенной платы
  i : integer;

  ...

  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

  ...

  // Работа с платой

  // Проверка настроек интерфейса (EnableDigExtStartCheckBox – параметр
отвечающий за выбор режима внешнего запуска по цифровому сигналу)

  if EnableDigExtStartCheckBox = true then
    XDSPDigExtStartEnable(device); // включение режима внешнего запуска по
    цифровому сигналу
  else
    XDSPDigExtStartDisable(device); // выключение режима внешнего запуска по
    цифровому сигналу
  end;

  ...

  // Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XdspSetProcessLength - Установить количество отсчетов дискретизации и длину DMA транзакции

Декларация: **Function XdspSetProcessLength(hDeviceHandle:TDeviceHandle;
dwLength:DWORD):DWORD**

Функция предназначена для установки длины (в отсчетах) процесса дискретизации и DMA транзакции. По запуску дискретизации плата осуществляет оцифровку входных сигналов и заполнение внутренней буферной памяти заданным этой функцией количеством отсчетов. Запуск DMA транзакции активизирует процесс копирования этого диапазона внутренней памяти в буфер компьютера.

Значение задаваемое этой функцией не должно превышать 8 М ($8 \cdot 2^{20} = 8388608$).

Параметры функции: hDeviceHandle - дескриптор открытой платы;
dwLength - длина.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Const
  TransactionLength = 262144; //1 М отсчетов на канал

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // указатель на номер подключенной платы
  i : integer;

...

// Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
  @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
  @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
  @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
  @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

// Устанавливаем длину оцифровки и длину передаваемого DMA буфера 1М
(отсчетов)
XdspSetProcessLength(device,TransactionLength);

...

// Работа с платой

...

// Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
XdspCloseDevice(device);
```

Function XdspSetSampleFrequencyDivider - Установить делитель частоты процесса оцифровки

Декларация: **Function XdspSetSampleFrequencyDivider(hDeviceHandle:TDeviceHandle;
dwDivider:DWORD):DWORD**

Функция предназначена для установки коэффициента дискретизации. При делителе равном 1 плата производит оцифровку сигнала с максимальной частотой дискретизации — 100 МГц. Увеличивая этот параметр можно понижать частоту дискретизации. По умолчанию значение делителя частоты принимается равным 1.

Параметры функции: hDeviceHandle - дескриптор открытой платы;
dwDivider - делитель частоты дискретизации.

Примечание: Значение dwDivider должно находиться в диапазоне от 1 до 255.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // указатель на номер подключенной платы
  i : integer;

...

  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

  // Устанавливаем делитель частоты дискретизации равным 2 (частота
дискретизации при этом 50 МГц)
  XdspSetSampleFrequencyDivider(device,2);

...

  // Работа с платой

...

  // Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XdspAddrCntrToADCDataEnable – включение режима подмены данных АЦП счетчиками

Декларация: **FunctionXdspAddrCntrToADCDataEnable (hDeviceHandle: TdeviceHandle):
DWORD**

Функция предназначена для тестирования работы канала между платой АЦП и платой PSI-express. При этом отсекаются данные со входа АЦП и в канал поступает циклически генерируемая возрастающая последовательность (0-16383).

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функции в среде Delphi показан при описании функции *XdspAddrCntrToADCDataDisable* (см. следующую страницу).

Function XdspAddrCntrToADCDataDisable – выключение режима подмена данных АЦП счетчиками

Декларация: **Function XdspAddrCntrToADCDataDisable (hDeviceHandle:TDeviceHandle):
DWORD**

Функция предназначена для выключения режима тестирования работы канала между платой АЦП и платой PSI-express.

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функций *XdspAddrCntrToADCDataEnable* и *XdspAddrCntrToADCDataDisable* в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель разрешения подмены данных АЦП счетчиками
  AddrCntrTestCheckBox : boolean;
  // указатель на номер подключенной платы
  i : integer;
...

  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

  if AddrCntrTestCheckBox = true then begin

    // Включаем режим подмены данных АЦП адресными счетчиками.
    XdspAddrCntrToADCDataEnable(device);

  else

    // Выключаем режим подмены данных АЦП адресными счетчиками
    XdspAddrCntrToADCDataDisable(device);

  end;

  //Работа с платой
...

  // Закрываем плату перед завершением работы, номер подключенной платы опущен
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XDSPSetDAC1Default – запись в ЦАП1 значения по умолчанию

Декларация: **Function XDSPSetDAC1Default(hDeviceHandle:TDeviceHandle;
dwData:DWord):DWord**

Функция устанавливает значение сигнала для ЦАП1 которое генерируется при простое платы, т.е. когда процесс генерации сигнала не активен.

Параметры функции: hDeviceHandle – дескриптор открытой платы;
dwData – значение записываемое в ЦАП.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device      : TDeviceHandle;
  // Значение, записываемое в ЦАП по умолчанию
  dwData : Dword=8192;
  // указатель на номер подключенной платы
  i : integer;

...
// Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
  @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
  @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
  @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
  @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

...

// Настройка устройства

...

// Запись в ЦАП1 значения по умолчанию
XdspSetDAC1Default(device,dwData);

...

// Сброс платы
XdspResetAll(device);

// Работа с платой
...

// Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
XdspCloseDevice(device);
```

Function XDSPSetDAC2Default – запись в ЦАП2 значения по умолчанию

Декларация: **Function XDSPSetDAC2Default(hDeviceHandle:TDeviceHandle;
dwData:DWord):DWord**

Функция устанавливает значение сигнала для ЦАП2 которое генерируется при простое платы, т.е. когда процесс генерации сигнала не активен.

Параметры функции: hDeviceHandle – дескриптор открытой платы;
dwData – значение записываемое в ЦАП.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device      : TDeviceHandle;
  // Значение, записываемое в ЦАП по умолчанию
  dwData : Dword=8192;
  // указатель на номер подключенной платы
  i : integer;

...
// Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
  @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
  @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
  @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
  @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);

...

// Настройка устройства

...

// Запись в ЦАП2 значения по умолчанию
XdspSetDAC2Default(device,dwData);

...

// Сброс платы
XdspResetAll(device);

// Работа с платой
...

// Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
XdspCloseDevice(device);
```

Function XDSPEnableDMARead – разрешение чтения DMA транзакций

Декларация: **Function XDSPEnableDMARead(hDeviceHandle:TDeviceHandle):DWORD**

Функция включает PCI интерфейс для работы с платой в "Slave" режиме. При этом плата получает возможность производить чтение данных из оперативной памяти компьютера. Это необходимо для передачи сгенерированного компьютером сигнала в память ЦАП1 и ЦАП2 платы ADC-DAC-125_14.

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функции в среде Delphi показан при описании функции *XDSPDisableDMARead* (см. следующую страницу).

Function XDSPDisableDMARead – запрет чтения DMA транзакций

Декларация: **Function XDSPDisableDMARead(hDeviceHandle:TDeviceHandle):DWORD**

Функция возвращает PCI интерфейс в режим "*Master*" для работы с платой.

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TDeviceHandle;
  // Переменная для хранения результатов работы функции
  result : Dword;
  // указатель на номер подключенной платы
  i : integer;
...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
  ...

  // Работа с платой

  // Разрешение чтение DMA транзакции
  XDSPEnableDMARead(device);
  // Инициализирует плату
  XdspResetAll(device);

  // Запуск копирования данных с 3-го DMA буфера компьютера в 3-й буфер ЦАП-а на
плате
  // Время ожидания таймаута 20с
  result:=XdspDMATransactionStart(device,2,2,20000);

  // Если ошибка
  If result=XDSP_STATUS_ERROR Then Begin
    // Произошла ошибка при выполнении функции, код ошибки можно получить
используя GetLastError
  End;

  // Запрет чтения DMA транзакции
  XDSPDisableDMARead(device);

  // Аналогичная операция выполняется и по 4-му буферу DMA
...

  // Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XDSPHistBuffEnable – разрешить видене буфера предистории

Декларация: **Function XDSPHistBuffEnable(hDeviceHandle:TDeviceHandle):DWORD**

Функция включает работу буфера предистории на плате. При работе с платой в режиме внешнего запуска может возникнуть необходимость обработки данных предшествовавших сигналу внешнего запуска. Включение буфера предистории позволяет получить 1024 (из 16 М) отсчетов данных предшествовавших сигналу внешнего запуска, по каждому каналу. Размер буфера имеет фиксированную длину – 1024 отсчета.

Функция возвращает XDSP_STATUS_SUCCESS в случае успешного выполнения или XDSP_STATUS_ERROR (см. константы и типы для работы с API функциями библиотеки, описанные в файле "*xdspapi.pas*"), код ошибки можно получить используя стандартную Windows API функцию *GetLastError*.

Параметры функции: hDeviceHandle – дескриптор открытой платы

Пример использования функции в среде Delphi показан при описании функции *XDSPHistBuffDisable* (см. следующую страницу).

Function XDSPHistBuffDisable – запретить буфер истории

Декларация: **Function XDSPHistBuffDisable(hDeviceHandle:TDeviceHandle):DWORD**

Функция выключает работу буфера предыстории на плате.

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функций *XDSPHistBuffEnable* и *XDSPHistBuffDisable* в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель включения ведения истории
  HistBuffCheckBox : boolean;
  // Указатель количества включенных устройств
  dev_count : integer;
  // указатель на номер подключенной платы
  i : integer;

...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
  функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
...

  // Работа с платой
  // Подсчет числа включенных в систему устройств. Результат заносится в
  dev_count

  // Проверка настроек программы
  for j :=1 to dev_count do
    if HistBuffCheckBox = true then
      XDSPHistBuffEnable(device); // Включение режима ведения предыстории
    else
      XDSPHistBuffDisable(device); // Выключение режима ведения предыстории
    end;

...

  // Закрываем плату перед завершением работы, выбор номера устройства опущен
  (см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XDSPDACCounterEnable – включение счетчика к ЦАП-у

Декларация: **Function XDSPDACCounterEnable(hDeviceHandle:TDeviceHandle):DWORD**

Функция вместо сгенерированного сигнала подключает ко входам ЦАП на плате адресный счетчик. Режим используется для тестирования канала данных ЦАП. При таком режиме на выходе ЦАП-ов должен наблюдаться пилообразный сигнал.

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функции в среде Delphi показан при описании функции *XDSPDACCounterDisable* (см. следующую страницу).

Function XDSPDACCounterDisable – отключение счетчика от ЦАП-ов

Декларация: **Function XDSPDACCounterDisable(hDeviceHandle:TDeviceHandle):DWORD**

Функция отключает от входов ЦАП адресный счетчик, нормальный режим генерации сигнала.

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функций *XDSPDACCounterEnable* и *XDSPDACCounterDisable* в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель включения режима подключения счетчика ко входам ЦАП
  CheckBox5 : boolean;
  // Указатель количества включенных устройств
  dev_count : integer;
  // указатель на номер подключенной платы
  i : integer;

...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
...

  // Работа с платой
  // Подсчет числа включенных в систему устройств. Результат заносится в
dev_count

  // Проверка настроек программы
  for j :=1 to dev_count do
  if CheckBox5 = true then
    XDSPDACCounterEnable(device); // Подключение счетчика к ЦАП
  else
    XDSPDACCounterDisable(device); // Отключение счетчика от ЦАП
  end;

...

  // Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XDSPDACtoADCEnable – включение передачи данных с ЦАП на АЦП

Декларация: **Function XDSPDACtoADCEnable(hDeviceHandle:TDeviceHandle):DWORD**

Функция предназначена для тестирования памяти. Данная функция включает режим, при котором данные записанные в память ЦАП передаются в память АЦП на той же плате. А затем данные из памяти АЦП поступают на ПК. Таким образом проверяются основные каналы передачи данных.

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функции в среде Delphi показан при описании функции *XDSPDACtoADCDisable* (см. следующую страницу).

Function XDSPDACtoADCDisable - выключение передачи данных с ЦАП на АЦП

Декларация: **Function XDSPDACtoADCDisable(hDeviceHandle:TDeviceHandle):DWORD**

Данная функция выключает режим тестирования памяти описанный выше.

Параметры функции: hDeviceHandle – дескриптор открытой платы.

Пример использования функций *XDSPDACtoADCEnable* и *XDSPDACtoADCDisable* в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель включения режима подключения счетчика ко входам ЦАП
  CheckBox : boolean;
  // Указатель количества включенных устройств
  dev_count : integer;
  // указатель на номер подключенной платы
  i : integer;

...
// Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
  @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
  @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
  @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
  @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
...

// Работа с платой
// Подсчет числа включенных в систему устройств. Результат заносится в
dev_count

// Проверка настроек программы
for j:=1 to dev_count do
  if CheckBox = true then
    XDSPDACtoADCEnable(device); // Включение режима передачи данных с ЦАП на
АЦП
  else
    XDSPDACtoADCDisable(device); // Выключение режима передачи данных с ЦАП
на АЦП
  end;

...

// Закрываем плату перед завершением работы , выбор номера устройства опущен
(см. функцию XdspCloseDevice)
XdspCloseDevice(device);
```

Function XDSPWriteToADCDAC – запись управляющего слова в АЦП и ЦАП

Декларация: **Function XDSPWriteToADCDAC(hDeviceHandle:TDeviceHandle; data:Word; adc_flag:boolean; dac_flag:boolean):DWORD**

Функция по последовательному интерфейсу передает управляющее слово для АЦП и ЦАП. Это позволяет управлять тестовыми режимами АЦП (вкл/выкл паттерны), выполнять RESET ЦАП-ов и АЦП, и устанавливать уровень сигнала для ЦАПЗ.

Параметры функции: hDeviceHandle – дескриптор открытой платы;
data – записываемые данные;
adc_flag – флаг выбора АЦП;
dac_flag – флаг выбора ЦАП.

Примечание: функция в один момент времени может передавать данные только на АЦП или на ЦАП. Этот выбор делается при помощи флагов, выставляя соответствующий в состоянии «true».

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель количества включенных устройств
  dev_count : integer;
  // Указатель значения бита пересылаемого на устройство
  data,adc_reg : integer;
  // указатель на номер подключенной платы
  i : integer;

...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
...

  // Передача сигнала на ЦАП компаратора
  for k:=1 to dev_count do begin // Выбор подключенного устройства
    for j:=1 to 16 do begin // передача 16 бит управляющего слова
      if data = 0 then
        begin
          XdspWriteToADCDAC (device,0,false,true); // Передача 0
          XdspWriteToADCDAC (device,256,false,true);
        end
      else begin
          XdspWriteToADCDAC (device,1024,false,true); // Передача 1
          XdspWriteToADCDAC (device,1280,false,true);
        end;
      end;
    end;
  end;

  // Передача сигнала на АЦП
  for k:=1 to dev_count do begin // Выбор подключенного устройства
```



```
// Запись в adc_reg передаваемых данных
for j:=16 downto 1 do begin
    if adc_reg = 0 then
        XdspWriteToADC DAC(device,ADC_REG,true,false); // Передача 0
    else
        XdspWriteToADC DAC(device,ADC_REG,true,false); // Передача 1
    end;
end;
end;

...

// Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию xdspCloseDevice)
XdspCloseDevice(device);
```

Function XdspDACEnable — включение генерации сигнала ЦАП

Декларация: **Function XdspDACEnable(hDeviceHandle:TDeviceHandle):DWORD**

Функция включает режим генерации сигнала ЦАП.

Параметры функции: hDeviceHandle — дескриптор открытой платы.

Пример использования в среде Delphi показан при описании функции *XdspDACDisable* (см. следующую страницу).

Function XdspDACDisable — выключение генерации сигнала ЦАП

Декларация: **Function XdspDACDisable(hDeviceHandle:TDeviceHandle):DWORD**

Функция выключает генерацию сигнала ЦАП. Запускается через интерфейс программы командой «*EnableDACSize*».

Параметры функции: hDeviceHandle — дескриптор открытой платы.

Пример использования функций *XdspDACEnable* и *XdspDACDisable* в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель включения режима генерации сигнала ЦАП
  DACEnable : boolean;
  // Указатель количества включенных устройств
  dev_count : integer;
  // указатель на номер подключенной платы
  i : integer;

...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
  функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
...

  // Работа с платой
  // Подсчет числа включенных в систему устройств. Результат заносится в
  dev_count

  // Проверка настроек программы
  for j :=1 to dev_count do
  if DACEnable = true then
    XdspDACEnable(device); // Включение генерации сигнала ЦАП
  else
    XdspDACDisable(device); // Выключение генерации сигнала ЦАП
  end;

...

  // Закрываем плату перед завершением работы, выбор номера устройства опущен
  (см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XDSPReadDigInData — чтение цифровых сигналов

Декларация: **Function XDSPReadDigInData(hDeviceHandle:TDeviceHandle;
pdwData:PDWord):DWORD**

Функция выполняет чтение значений на цифровых выводах выбранного устройства.

Параметры функции: hDeviceHandle — дескриптор открытой платы;
pdwData — значение, прочитанное с цифровых выводов платы.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Указатель на значение цифровых сигналов
  temp : DWORD;
  // указатель на номер подключенной платы
  i : integer;

...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
...

  // Работа с платой

  // Чтение значений цифровых сигналов с платы
  XDSPReadDigInData(device,@temp);

  // Обработка полученных данных

...

  // Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Function XDSPWriteDigOutData — запись цифровых сигналов

Декларация: **Function XDSPWriteDigOutData(hDeviceHandle:TDeviceHandle;
dwData:DWord):DWord**

Функция выполняет запись, определенных пользователем значений, которые будут отображаться на цифровых выводах установленной платы.

Параметры функции: hDeviceHandle — дескриптор открытой платы;
dwData — значение, записываемое в цифровые выводы платы.

Пример использования функции в среде Delphi:

```
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

Var
  // Переменная для хранения дескриптора устройства
  device : TdeviceHandle;
  // Значение цифровых сигналов
  temp : DWord;
  // указатель на номер подключенной платы
  i : integer;

...
  // Открываем плату, проверка ошибок и выбор номера устройства опущены (см.
функцию XdspOpenDevice)
  device:=XdspOpenDevice((i-1), @XdspDevices[i].InterruptHandle,
    @XdspDevices[i].DmaBuffer1, @XdspDevices[i].DmaBuffer2,
    @XdspDevices[i].DmaBuffer3, @XdspDevices[i].DmaBuffer4,
    @XdspDevices[i].DmaBufferPhysAddr1, @XdspDevices[i].DmaBufferPhysAddr2,
    @XdspDevices[i].DmaBufferPhysAddr3, @XdspDevices[i].DmaBufferPhysAddr4);
...

  // Работа с платой

...

  // Читаем из интерфейса программы значения для цифровых сигналов, записывая их
в переменную temp

  // Запись значений цифровых сигналов в плату
  XDSPWriteDigOutData(device, temp);

...

  // Работа с платой

...

  // Закрываем плату перед завершением работы, выбор номера устройства опущен
(см. функцию XdspCloseDevice)
  XdspCloseDevice(device);
```

Рекомендованный алгоритм работы с платой

```
// Проверка ошибок опущена.
// Подключаем модуль с описанием констант, типов и импортируемых функций
Uses xdspapi;

...

Const ADCBufLen = 16777215; полный размер 1-го буфера АЦП, 8М двойных слов

Var
  // Переменные для хранения дескриптора устройства
  device:TDeviceHandle;
  hdv : THandle;
  // Переменная для хранения результата работы функций
  result : DWord;
  // Указатели на DMA буфер, тип PDmaMemoryBuffer определен как указатель на массив
  // Указатель прерывания от подключенного устройства
  device_interrupt : THandle;
  // из 16 М слов (максимальный размер буфера пересылаемого DMA транзакцией)
  dma_buffer1 : PDmaMemoryBuffer;
  dma_buffer2 : PDmaMemoryBuffer;
  dma_buffer3 : PDmaMemoryBuffer;
  dma_buffer4 : PDmaMemoryBuffer;
  // Указатель на физический адрес DMA буфера, тип Dword
  dma_buffer_phys1 : Dword;
  dma_buffer_phys2 : Dword;
  dma_buffer_phys3 : Dword;
  dma_buffer_phys4 : Dword;

  Chanel1 : Array[0..ADCBufLen] of SmallInt; //буфер для оцифрованного сигнала 1
канала
  Chanel2 : Array[0..ADCBufLen] of SmallInt; //буфер для оцифрованного сигнала 2
канала

  // значение, записываемое в ЦАП по умолчанию
  DACdefault : integer = 8192;
  // указатель на номер открытой платы
  k : integer;

Begin

  for k:=1 to 4 do begin
    // Открытие устройства и инициализация (в программе принято число
устройств равное 4). Получение указателей на 4-ре буфера памяти.
    hdv:=XdspOpenDevice((k-1), @XdspDevices[k].InterruptHandle,
@XdspDevices[k].DmaBuffer1, @XdspDevices[k].DmaBuffer2,
@XdspDevices[k].DmaBuffer3, @XdspDevices[k].DmaBuffer4,
@XdspDevices[k].DmaBufferPhysAddr1, @XdspDevices[k].DmaBufferPhysAddr2,
@XdspDevices[k].DmaBufferPhysAddr3,@XdspDevices[k].DmaBufferPhysAddr4);
    // нумерация устройств в ПК 0-3

    device(k):=hdv;

    // сброс и инсталяция платы
    XdspResetAll(device(k));

    // запись в ЦАП1 и ЦАП2 значений по умолчанию
    XdspSetDAC1Default(device(k),DACdefault);
    XdspSetDAC2Default(device(k),DACdefault);

    // сброс и инсталяция платы
```



```
...  
  
// разбор данных по каналам полученных из АЦП  
for i:=0 to ADCBufLen do Begin  
    // массив DMA буфера начинается с индекса 1  
    Chanel1[i]:=SmallInt(dma_buffer1^[i+1]);  
    Chanel2[i]:=SmallInt(dma_buffer2^[i+1]);  
End;  
  
// дальнейшая обработка полученных данных  
  
...  
...  
...  
  
// Закрытие устройства  
XdspCloseDevice(device(k));  
  
...  
end;  
  
End.
```


Описание сигнальных выводов платы

Цифровые выводы платы

На плате установлено два разъема цифровых сигналов J10 и J12

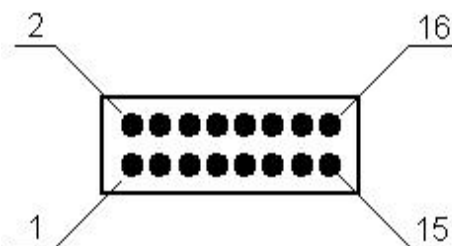


Рис. 12. Разъем цифровых сигналов AMP IPC2-16M

В таблице 9 приведены наименование цифровых выводов разъема AMP IPC2-16M

Таблица 9. Наименование выводов разъемов AMP IPC2-16M

Обозначение разъема на плате	Имя сигнала	Назначение	Номер контакта	Номер контакта GND
J12	FPGA-IN0	Контакты не используются, могут назначаться по требованию заказчика	1	2
J12	FPGA-IN1		3	4
J12	FPGA-IN2		5	6
J12	FPGA-OUT0	Инвертированное значение флага оцифровки данных («временное окно»)	7	8
J12	FPGA-OUT1	Значение флага оцифровки данных («временное окно»)	9	10
J12	FPGA-OUT2	Сигнал внешнего запуска оцифровки данных	11	12
J12	3,3V	Питание 3,3 V *	13,15	14,16
J10	FPGA-IN3	Контакты не используются, могут назначаться по требованию заказчика	1	2
J10	FPGA-IN4		3	4
J10	FPGA-IN5	Цифровой сигнал внешнего запуска	5	6
J10	FPGA-OUT3	Инвертированное значение сигнала внешнего запуска оцифровки данных	7	8
J10	FPGA-OUT4	Сигнал начала DMA-транзакции	9	10
J10	FPGA-OUT5	Инвертированное значение сигнала начала DMA-транзакции	11	12
J10	3,3V	Питание 3,3 V *	13,15	14,16

* В случае подключения аппаратуры с питанием 3,3 В на плату необходимо установить два резистора R114 и R98 номиналом 0 Ом (перемычка).

Примечание: все данные, передаваемые или получаемые с цифровых выводов, подвергаются аппаратному инвертированию (лог. «0» на цифровом разъеме соответствует лог. «1» на ПК).

После старта процесса транзакции сигнал FPGA-OUT4 устанавливается в состояние лог. «1» (FPGA-OUT5 в состоянии лог. «0»).

Если режим внешнего запуска выключен, то оцифровка данных начинается автоматически.

При этом сигнал FPGA-OUT1 переходит в состояние лог. «1» (FPGA-OUT0 в состоянии лог. «0»), сигнал FPGA-OUT4 переходит в состояние лог. «0», (FPGA-OUT5 в состоянии лог. «1»).

В случае, если режим внешнего запуска включен, то система ожидает появления сигнала внешнего запуска. При этом сигнал FPGA-OUT4 находится в состоянии лог «1» (FPGA-OUT5 в состоянии лог. «0»). Как только появляется сигнал внешнего запуска, FPGA-OUT2 в состоянии лог. «1» (FPGA-OUT3 в состоянии лог. «0»), начинается процесс оцифровки данных, что сигнализируется переходом сигнала FPGA-OUT1 в состояние лог. «1» (FPGA-OUT0 в состоянии лог. «0»), сигнал FPGA-OUT4 в состояние лог. «0» (FPGA-OUT5 в состоянии лог. «1»).

Примечание: в скобках указаны соответствующие инвертированные цифровые сигналы.

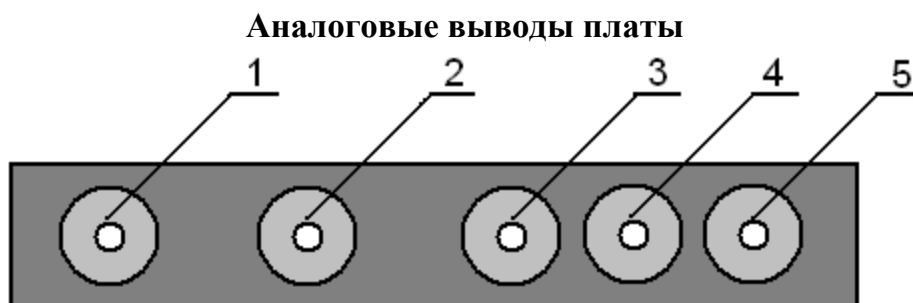


Рис. 13. Аналоговые выводы платы SMB

Тип разъема, используемый для аналоговых вводов — SMB.

В таблице 10 укажем назначение аналоговых выводов платы.

Таблица 10. Аналоговые выводы платы

Номер вывода	Имя сигнала	Назначение
1	Сигнал АЦП1	Входы аналоговых сигналов на преобразователи АЦП
2	Сигнал АЦП2	
3	Сигнал ЦАП1	Выходы аналоговых сигналов с преобразователей ЦАП
4	Сигнал ЦАП2	
5	Сигнал компаратора	Вход аналогового сигнала на компаратор

Компаратор, установленный на плате, используется для формирования сигнала внешнего аналогового запуска. Уровень срабатывания компаратора задается цифро-аналоговым преобразователем (DAC3). На вход компаратора поступает сигнал с выхода преобразователя DAC3 и аналоговый сигнал 5 (рис. 13).

На основе сигнала 5 (рис. 13) формируется импульсный сигнал с фронтами в точках пересечения сигнала 5 (синусоидальная форма) с опорным напряжением на выходе преобразователя DAC3.

Меняя напряжение на выходе DAC3, меняем и уровень срабатывания компаратора, что приводит к изменению длительности лог. «1» и лог. «0» (период импульсного сигнала при этом остается неизменным), а, следовательно, и время срабатывания от переднего или от заднего фронта. Момент срабатывания зависит от сделанных пользователем настроек в поставляемой с платой программном обеспечении «WIN_DO_125AS5_Oscilloscope» (вкладка *View*, компонент *ExternalPuskEdgeNegative*).

Контакты:

127247, Москва, Дмитровское ш. 100, стр. 3, ЗАО «Компания Сигнал»
Тел. (495) 788-40-67
Факс (905) 708-45-64
Skype: signal488
E-mail: signal@signal.ru
www.signal.ru