

# ADC-1293G

плата ввода/вывода аналоговых и дискретных сигналов  
с гальванической развязкой  
для персональных IBM совместимых компьютеров

## *Руководство пользователя*



**«ХОЛИТ Дэйта Системс, Лтд»**

✉ 03056, Украина, Киев-56,  
ул. Политехническая 16, уч.корп.12,  
к.019

(044) 241-8739, 241-67-54, 492-31-08, 491-31-09

support@holit.com.ua

info@holit.com.ua

www.holit.com.ua



**ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ** **4**

**КОНФИГУРАЦИЯ** **5**

- монтажная схема
- установка коннекторов
- схема разъема
- порядок установки
- подключение внешних устройств

**АДРЕСНОЕ ПРОСТРАНСТВО и ФОРМАТЫ ДАННЫХ** **9**

- управление платой
- карта адресов
- описание регистров
- режим внешнего запуска
- команды управления

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ** **12**

- библиотека подпрограмм и языки «высокого уровня»
- форматы данных
- модели памяти
- описание библиотеки функций DOS
- описание библиотеки функций Windows
- описание библиотеки функций LabVIEW

Плата ADC-1293G предназначена для преобразования аналоговых сигналов в цифровую форму, а также для ввода/вывода дискретных ТТЛ сигналов.

Плата выполнена в стандарте плат IBM PC AT и устанавливается в любой из свободных разъемов, расположенных непосредственно в компьютере. Подключение к внешним устройствам осуществляется через разъем платы со стороны задней панели компьютера.

ADC-1293G является функционально полным комплексом, включающим в себя многоканальный 12-и разрядный аналого-цифровой преобразователь с частотой преобразования до 70 кГц на один канал и 13 ТТЛШ входов/выходов.

Аналого-цифровой преобразователь работает в двух режимах подключения источников внешних сигналов: режим 1 - 16 дифференциальных каналов, режим 2 - 32 канала относительно общей земли.

Внутренняя синхронизация и работа в режиме реального времени обеспечивается RISC-микроконтроллером, при этом управление платой выполняется также просто, как запись/чтение 8-битового слова в любой из портов В/В IBM PC.

Комплект поставки:

1. Плата ADC-1293G.
2. Ответная часть разъема для подключения сигналов.
3. Дискета с программным обеспечением и руководством пользователя:

*adc1293g.doc* - текст данного руководства

*\PASCAL* - директория, содержащая библиотеку функций для языка 'Паскаль' и программы с примерами их вызова

*adc1293g.asm* - библиотека ассемблерных драйверов для платы ADC-1293G, адаптированная под язык 'Паскаль'

*adc1293g.obj* - объектный файл

*def1293g.inc* - файл определений

*demo1293.exe*, *graf1293.exe*, *intr1293.exe* - программы с примерами использования ассемблерных функций из библиотеки *ada1293g.asm*

*demo1293.pas*, *graf1293.pas*, *intr1293.pas* - исходные тексты демонстрационных программ

*\C* - директория, содержащая библиотеку драйверов для языка 'C'

*adc1293g.asm* - библиотека ассемблерных драйверов для платы ADC-1293G, адаптированная под язык 'C'

*adc1293g.obj* - объектный файл

*def1293g.h* - файл определений

*\WINLIB95* - директория, содержащая DLL-библиотеку и примеры ее применения в Delphi 3.0 и LabView 5.1.

Внимание !

Перед подключением каналов аналого-цифрового преобразователя мы настоятельно рекомендуем Вам изучить раздел руководства, в котором описывается подключение источников сигналов. Как показывает опыт, более 80 % проблем, возникающих при эксплуатации платы, связаны с неправильным подключением источников сигналов.

Плата устанавливается в компьютер в соответствии с описанием.

Распаивается ответная часть разъема для подключения сигналов в соответствии с руководством (особое внимание необходимо уделить заземлению сигналов).

Ответная часть разъема подсоединяется к плате.

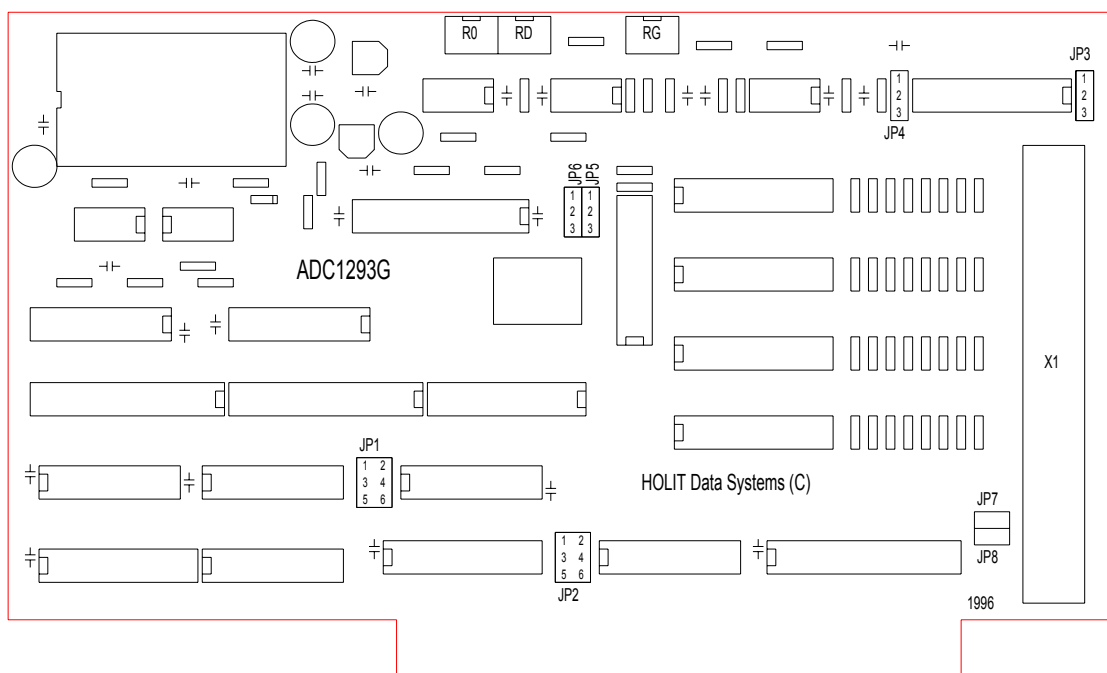
Включается компьютер.

<b>АЦП:</b>	количество аналоговых каналов	16 дифференциальных (!) или 32 с общей землей
	входное сопротивление	2 МОм
	входной диапазон	+/-10,24 В (+/-5,12 В)*
	разрядность АЦП	12 бит
	интегральная нелинейность	1МЗР
	дифференциальная нелинейность	1МЗР
	режим запуска	программный от внутреннего таймера внешний*
	частота дискретизации	70кГц
<b>ТТЛ-совместимые входы/выходы:</b>	число входов, не более	7 (! 3)*
	число выходов, не более	4 (! 3)*
	нагрузочная способность	5мА (макс. 25мА)
Режим ввода/вывода:	программный, по прерыванию	
Кварцевая стабилизация частоты дискретизации		
Гальваническая развязка	500В	
Система генерации прерываний: IRQ3(!), IRQ4, IRQ7, IRQ2(9)		
Внешний разъем платы:	32 входных аналоговых линий 11 линий дискретного ввода/вывода 15 линий земли 4 линии питания +/-15 В.	
Внешние факторы:	рабочая температура от +5 до +60 град. С температура хранения от -20 до +70 град.С относительная влажность от 5 до 90%	

---

(!) - базовая конфигурация, \* - уточняется при заказе

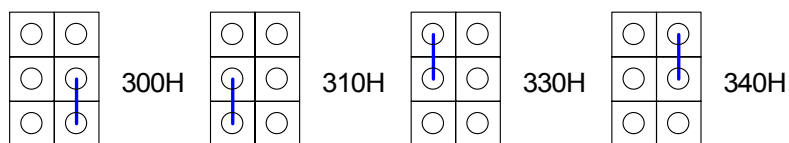
## МОНТАЖНАЯ СХЕМА



## УСТАНОВКА КОННЕКТОРОВ

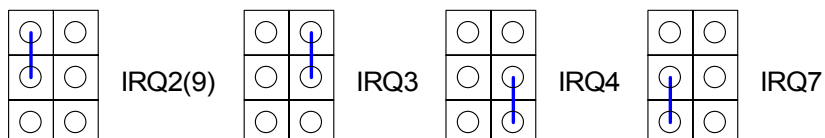
### Установка базового адреса

Адресные коннекторы определяют адрес платы в адресном пространстве компьютера. Плата занимает 16 последовательных ячеек начиная с базового адреса, устанавливаемого с помощью коннекторов JP1. Скорее всего, переключение базового адреса Вам потребуется при установке более чем одной платы в компьютер.



### Установка номера линии прерывания

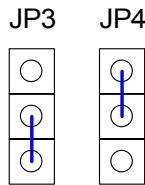
Плата может генерировать прерывания IRQ2(9), IRQ3, IRQ4, IRQ7 в зависимости от состояния коннектора JP2.



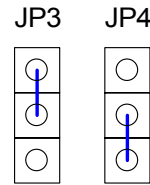
### Установка режима подключения внешних источников сигнала

Аналого-цифровой преобразователь работает в двух режимах подключения источника внешнего сигнала: 16 дифференциальных каналов или 32 канала относительно общей земли; который определяется состоянием коннекторов JP3 и JP4.

16 дифференциальных каналов



32 канала относительно общей земли



### Конфигурация линий дискретного ввода/вывода

Установка режима выполняется изготовителем по согласованию с заказчиком с помощью коннекторов JP5 и JP6. Базовая конфигурация предполагает установку линий SW4, MX2, MX3 на вывод; линий Din0, Din1, Din2 на ввод.

**Разрешение вывода питания +-15В** на разъем выполняется установкой коннекторов JP7 и JP8.

### Заводская установка коннекторов

Базовый адрес : 300H

Установлен режим : 16 дифференциальных каналов, 3 вх., 3 вых. цифровые линии

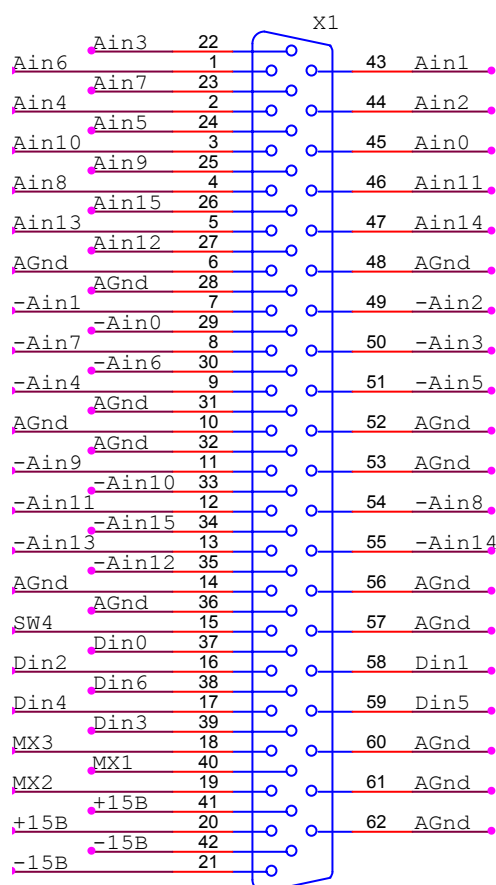
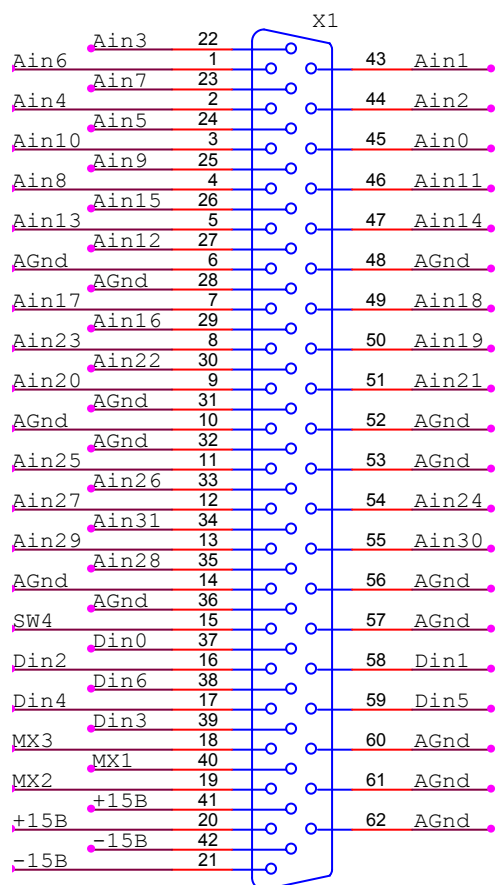
Номер линии прерывания: не установлен

Питание +-15В на разъем не выведено

## СХЕМА РАЗЪЕМА

режим 32 канала с общей землей

режим 16 дифференциальных каналов



Ain0...Ain31  
Ain0...Ain15  
-Ain0...-Ain15  
Din0..Din6  
MX1..3(\*),SW4  
+EE,-EE  
GND

аналоговые входы в режиме с общей землей  
неинвертирующие входы при диф.подключении  
инвертирующие входы при диф.подключении  
дискретные линии ввода  
дискретные линии вывода  
линии питания +-15В  
общий

(\* ) MX3 может быть настроена как линия внешнего запуска АЦП.

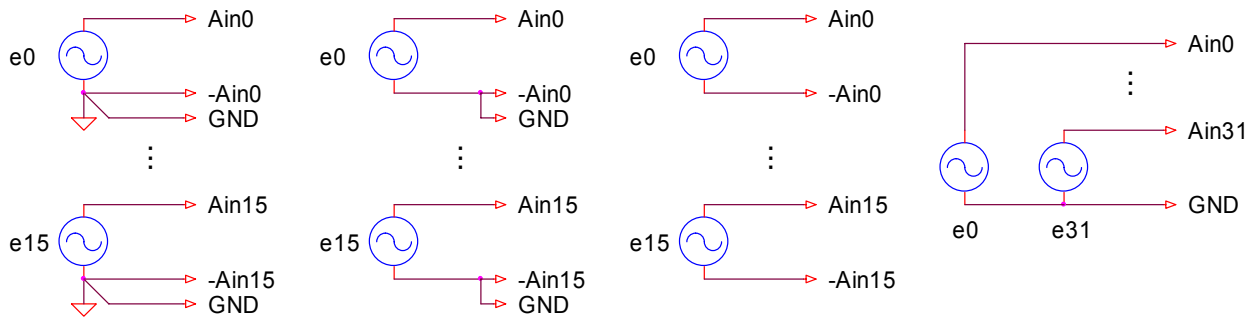
## ПОДКЛЮЧЕНИЕ ВНЕШНИХ УСТРОЙСТВ

Перед подключением каких-либо источников сигнала необходимо обеспечить общий контур заземления платы и подключаемых к нему приборов. Для этого необходимо соединить один из контактов, поименованный GND, разъема платы с контуром заземления Ваших приборов.

При использовании платы в режиме 16 дифференциальных каналов каждый источник сигнала подключается к соответствующему каналу АЦП Д ВУМЯ проводами. Неинвертирующий вход АЦП подключается к выходной клемме источника, а инвертирующий вход АЦП заземляется непосредственно на корпусе источника сигнала.

Вариант использования платы в 32-канальном режиме с общим проводом рассчитан на подключение к единому блоку источников сигналов с общей землей.

Примеры подключения:



**ВНИМАНИЕ!** Отсутствие общего контура заземления может привести к поломке платы.

## ПОРЯДОК УСТАНОВКИ

1. Проверьте упаковку и компоненты на отсутствие механических повреждений.
2. Установите необходимые Вам коннекторы на плате.
3. Выключите питание компьютера.
4. Снимите крышку с компьютера в соответствии с его описанием
5. Вывинтите крепежный винт заглушки одного из свободных слотов внутри компьютера.  
*Желательно, для уменьшения помех, выбирать слот, наиболее удаленный от блока питания компьютера и от карты адаптера дисплея*
6. Установите плату в свободный разъем и закрепите ее винтом
7. Закройте крышку компьютера

## Характерные неисправности и методы их исправления

Неисправность, внешнее проявление	Вероятная причина	Метод устранения
Отсутствие сигнала на канале АЦП	Неправильное подключение источника сигнала к разъему платы	Подключить сигнал в соответствии с руководством Выбранный базовый адрес не соответствует программной установке
Повышенный уровень шума	Неправильное заземление Неверный номер канала АЦП Неподключенный канал	Обеспечить заземление в соответствии с руководством

**! Неиспользуемые аналоговые входы следует заземлять.**

В случае, если не удастся избавиться от неисправности описанными методами, необходимо сообщить об этом в фирму - изготовитель.



## УПРАВЛЕНИЕ ПЛАТОЙ

Управление платой осуществляется через порты ввода/вывода с автоматическими устанавливаемыми и сбрасываемыми битами готовности. Управление аналоговым вводом и цифровым вводом/выводом, частотой преобразования и т.п. осуществляется RISC-микроконтроллером PIC16CXXX (MicroChip). Компьютеру доступны только регистр команд и регистры данных, через которые идет обмен между компьютером и RISC-микроконтроллером. Предусмотрен также аппаратный сброс микроконтроллера через порт ввода/вывода в случае его «зависания».

## КАРТА АДРЕСОВ

Плата занимает 16 последовательных ячеек в пространстве ввода/вывода компьютера, начиная с базового адреса BASE. Обращения к портам платы являются обнбайтными.

Адрес порта	Направление	Описание
BASE+0	запись 8 бит	регистр команд
BASE+0	чтение 8 бит	регистр младшего байта данных
BASE+1	запись 8 бит	аппаратный сброс микроконтроллера
BASE+1	чтение 8 бит	регистр старшего байта данных, битов готовности и бита-индикатора наличия платы
BASE+2 .. BASE+F	-	зарезервированы

## ОПИСАНИЕ РЕГИСТРОВ

### Регистр младшего байта данных

Бит	D7	D6	D5	D4	D3	D2	D1	D0
Имя	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0

DAT0..DAT7 - младшие разряды 12-битного кода АЦП  
 DAT0..DAT7 - состояние дискретных входных линий Din0..Din6,  
 при этом состояние DAT7 неопределено.

### Регистр старшего байта данных и битов готовности

Бит	D7	D6	D5	D4	D3	D2	D1	D0
Имя	DETECT	RDYin	RDYout	X	DAT11	DAT10	DAT9	DAT8

DETECT - бит-индикатор наличия платы:  
 0 - плата установлена, 1 - плата отсутствует  
 RDYin - бит готовности регистра команд  
 0 - плата готова принимать команду к выполнению  
 RDYout - бит готовности регистров данных  
 0 - регистры данных содержат результат аналого-цифрового преобразования или состояние дискретных линий ввода Din0..Din6  
 X - состояние неопределено  
 DAT11..DAT8 - старшие разряды 12-битового кода АЦП

Результат аналого-цифрового преобразования представлен в формате двенадцатибитного несмещенного целого со знаком:

Преобразуемое напряжение	Значение кода результата
+FSR/2 - 1LSB	от 01111111110 до 01111111111
+FSR/2 - 2LSB	от 01111111101 до 01111111110
+FSR/2 - 3LSB	от 01111111100 до 01111111101
...	...
AGND + 1LSB	от 00000000000 до 00000000001
AGND (0)	от 11111111111 до 00000000000
AGND - 1LSB	от 11111111110 до 11111111111
...	...
-FSR/2 + 3LSB	от 10000000010 до 10000000011
-FSR/2 + 2LSB	от 10000000001 до 10000000010
-FSR/2 + 1LSB	от 10000000000 до 10000000001

FSR (Full Scale Range) - диапазон преобразуемых напряжений, 1LSB = FSR/4096  
AGND - вход АЦП соединен с аналоговой "землей"

Пример получения стандартного целочисленного значения смотрите в файле adc1293g.asm.

#### Регистр команд

Бит	D7	D6	D5	D4	D3	D2	D1	D0
Назначение	Команда				Параметры *			

\* - если параметры не требуются биты D0..D4 должны быть равны 0

#### Регистр сброса

Операция записи в данный регистр инициирует аппаратный сброс управляющего микроконтроллера. Используется только в случае возникновения сбоя и отсутствии связи с микроконтроллером.

### РЕЖИМ ВНЕШНЕГО ЗАПУСКА

Установка режима внешнего запуска АЦП осуществляется посылкой команды cmdCfg с установленным пятым битом ( 1001wxxx ). После установки режима внешнего запуска работа с АЦП возможна только по внешнему сигналу на линии MX3. Запуск АЦП по любой команде будет синхронизирован с сигналами на этой линии. Запускающий импульс - положительный, длительностью не менее 0.7 мкс.

По команде cmdADChen запуск АЦП произойдет только по приходу запускающего импульса. Для следующего запуска необходимо снова подать команду cmdADChen.

По команде cmdFlowADC после прихода запускающего импульса начнется обработка АЦП по каналам, ранее заданной командой cmdSetSeqChnls, с межканальной частотой опроса, ранее заданной командой cmdSetRate. Опрос будет производиться циклически, для его завершения необходимо после последнего принятия данных немедленно послать пустую команду (0) в регистр команд! Для следующего цикла опроса по внешнему сигналу необходимо снова подать команду cmdADChen.

## КОМАНДЫ УПРАВЛЕНИЯ

<b>00000000</b> (cmdEmpty)	пустая команда. Используется в специальных случаях (см. команду cmdFlowADC)
<b>100ewxxx</b> (cmdCfg)	задает конфигурацию аналоговых и цифровых линий e - режим запуска АЦП (0 – программный запуск, 1- внешний запуск по линии MX3) w - тип подключения аналоговых линий (0 – однопроводные, 1 – дифференциальные) xxx - число входных цифровых линий (1, 3 или 7)
<b>111xxxxx</b> (cmdADChen)	одно преобразование АЦП по каналу , заданному параметром 'xxxxx'. Результат возвращается в регистр данных. Обработка данной команды происходит за 30 - 40 мкс. xxxxx - номер канала 0..15 / 0..31
<b>01100000</b> (cmdSetSeqChnls)	установка последовательности опроса каналов АЦП. После отправки в порт данной команды необходимо поочередно выдать номера каналов в тот же порт (не более 32-х). После выдачи последнего номера необходимо послать 0FFh.
<b>10100000</b> (cmdSetRate)	установка частоты опроса каналов при использовании команды cmdFlowADC, заданных командой cmdSetSeqChnls. После выдачи в порт данной команды необходимо поочередно выдать сначала младший байт, затем старший байт слова, содержащего значение задержки (Тз) в мкс. Период опроса вычисляется как (Тз + 14) мкс. Т.о. максимальная частота достигается при Тз=0 и равна $1/(Тз+14) = 1/(0+14) = 71.4 \text{ КГц} .$
<b>00100000</b> (cmdFlowADC)	запуск АЦП по каналам, ранее заданным командой cmdSetSeqChnls, с межканальной частотой опроса, ранее заданной командой cmdSetRate. Опрос производится циклически, для его завершения необходимо после последнего принятия данных немедленно послать пустую команду (0) в регистр команд!
<b>0100xxxx</b> (cmdDOut)	установка внешних цифровых линий xxxx - код установки внешних линий MX3, MX2, MX1, SW4
<b>10000000</b> (cmdDIn)	опрос состояния внешних цифровых линий. Последующее чтение младшего регистра данных возвратит состояние внешних цифровых линий.

Данный раздел описания предназначен для программистов, собирающихся писать свои собственные программы для работы с платой. В качестве базовых языков были выбраны языки 'Си' и 'Паскаль', поскольку они являются одними из самых широко распространенных и применяемых языков и практически все современные языки, включая 'Бейсик' и 'Фортран', имеют возможность вызывать функции, написанные в стандарте 'Си' или 'Паскаль'. Для приобретенного Вами модуля фирма поставляет готовую библиотеку подпрограмм (драйвер), в которую мы попытались включить множество разнообразных функций для облегчения программирования. Готовая библиотека подпрограмм позволяет Вам использовать практически все возможности платы, не вдаваясь в тонкости программирования на уровне ассемблера и портов ввода - вывода. В том случае, если Вы все же собираетесь сами программировать плату через порты ввода-вывода, то наша библиотека может быть использована Вами в качестве законченного и отлаженного примера, на основе которого Вы можете реализовать свои алгоритмы. Мы надеемся, что описываемая ниже библиотека упростит и ускорит написание Ваших программ.

## БИБЛИОТЕКА ПОДПРОГРАММ и ЯЗЫКИ «ВЫСОКОГО УРОВНЯ»

Языки 'Паскаль' и 'Си' позволяют без лишних сложностей использовать функции, написанные на языке ассемблер. Библиотека функций написана на языке 'Турбо Ассемблер' версии 3.2 (TASM), исходный текст Вы можете найти в файле adc1293g.asm, а откомпилированный вариант библиотеки при помощи 'Турбо Ассемблер' в файле adc1293g.obj (командная строка: tasm.exe /z/q/zn/w0/m2/mx \*.asm).

Законченный пример вызова функций библиотеки содержится в директории \PASCAL\\*.\*.

Для вызова функций из языка 'Си' Вам необходимо: создать файл проектов для языка 'Си' (xxxxxxx.pj) добавить в него файл adc1293g.obj (или adc1293g.asm) создать и добавить в него Ваш файл с будущей программой (например, example.c) добавить в начало файла строку #include "def1293g.h" (при этом файл def1293g.h, содержащий описания функций библиотеки должен находиться в текущей директории, в которой Вы создали файл проектов).

Теперь Вы можете писать свою программу и в любом месте вызывать функции из библиотеки adc1293g.obj. Если Вы пользуетесь средой Borland, то у Вас есть возможность включить в файл проектов не откомпилированную библиотеку, а ее исходный текст adc1293g.asm, при этом Вам необходимо в меню 'Option\Transfer' описать 'Турбо Ассемблер', добавив строку "TASM" в 'Program Path', " /mx /zi \$tasm" в 'Command Line' и пометить меню 'Translator'. После этого Вы можете в проектном файле, нажав <Ctrl O> для файла adc1293g.asm, установить режим компилирования при помощи 'Турбо Ассемблер'.

Простой пример: осуществим асинхронный ввод с первого (0) канала АЦП

```
[] 'Паскаль':
{$L adc1293g.obj}
VAR adCode : integer;
{$F+}
PROCEDURE SetADChan1293( chan : word ); EXTERNAL;
FUNCTION AD1293 : integer; EXTERNAL;
{$F-}
BEGIN
  SetADChan1293( 0 ); { установим канал АЦП }
  adCode:= AD1293; { считываем код с АЦП }
END.
```

```
[] 'Си':
#include "def1293g.h"
main()
{
  int adCode;
```

```
SetADChan1293( 0 ); // установим канал АЦП
adCode = AD1293(); // считываем код с АЦП
}
```

## ФОРМАТЫ ДАННЫХ

Данные, введенные с АЦП при помощи функций драйвера adc1293g.asm, связаны с напряжением следующей таблицей (соответствие кода АЦП напряжению):

Код	Напряжение	Пример
2047	MAX	+10.24 В
0	0	0.0 В
-2048	MIN	-10.24 В

MAX - максимальное напряжение (зависит от установленных переключателей конфигурирования входного диапазона).

MIN - минимальное напряжение (зависит от установленных переключателей конфигурирования входного диапазона).

## МОДЕЛИ ПАМЯТИ

Все функции библиотеки предполагают использование дальних указателей типа FAR, как для вызова процедур, так и для передачи адресов данных, поэтому компиляцию программы в языке 'Си' желательно проводить в модели памяти LARGE, в которой по умолчанию используются дальние вызовы процедур и дальние указатели на данные. В директории.\C находится драйвер, адаптированный под язык 'Си'.

Для использования драйвера adc1293g.asm в языке 'Си' необходимо использовать директиву ".MODEL LARGE, C". Для использования драйвера в языке 'Паскаль' необходимо использовать директиву ".MODEL LARGE, PASCAL".

## ОПИСАНИЕ БИБЛИОТЕКИ ФУНКЦИЙ DOS

### ФУНКЦИИ ОБЩЕГО НАЗНАЧЕНИЯ

Функции общего назначения предназначены для уведомления драйвера об установленной конфигурации платы с помощью коннекторов. При помощи коннекторов Вы можете изменить базовый адрес платы в пространстве ввода-вывода компьютера для работы с более чем одной платой. По умолчанию драйвер настроен на базовый адрес 300h. При его изменении Вам следует сообщить о них драйверу, вызвав описываемые ниже функции с соответствующими параметрами. После установки базового адреса (SetBaseAddress) и проверки наличия платы (Test) необходимо проинициализировать плату функцией Init.

#### **void SetBaseAddress( ulnt16 address )**

устанавливает новый базовый адрес в пространстве РС для функций данного драйвера (переменная 'address' должна быть равна 0x300, 0x310, 0x330, 0x340 и соответствовать установленному при помощи коннекторов адресу).

#### **ulnt16 Test( void )**

возвращает нулевое значение (0) в случае успешного тестирования присутствия платы и ненулевое значение (1) в противном случае. Ненулевое значение означает, что либо плата отсутствует в компьютере, либо функции SetBaseAddress был передан базовый адрес, не соответствующий установленному на плате.

#### **ulInt16 Init( ulInt16 ext, ulInt16 wire, ulInt16 digit )**

инициализация платы: 'ext' - устанавливает режим запуска АЦП (0 - программный запуск, 1 - внешний запуск), 'wire' - устанавливает тип подключения аналоговых линий (0 - однопроводные, 1 - дифференциальные), 'digit' - устанавливает число входных цифровых линий (1,3 или 7). По умолчанию плата инициализируется следующим образом : ext = 0, wire = 1, digit = 3. При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **void IRQ( ulInt16 on\_off )**

разрешает ('on\_off' = 1) или запрещает ('on\_off' = 0) прерывания во время работы ф-ций ADChanMulti, ADChansMulti... При работе на высоких частотах прерывания должны быть запрещены во избежание пропуска данных.

#### **void HardReset( void )**

инициирует аппаратный сброс управляющего микроконтроллера. Используется только в специальных случаях. После вызова этой функции необходимо заново проинициализировать плату функцией Init().

### **ФУНКЦИИ ПО РАБОТЕ С АНАЛОГО-ЦИФРОВЫМИ КАНАЛАМИ**

#### **void SetADChan( ulInt16 chan )**

устанавливает канал 'chan' для последующего ввода.

#### **int16 AD( void )**

запускает АЦП по ранее заданному каналу и возвращает результат преобразования. При обнаружении таймаута возвращает код 7FFFh(32767).

#### **int16 ADChan( ulInt16 chan )**

запускает АЦП по каналу 'chan' и возвращает результат преобразования. При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **ulInt16 ADChans( int16 \*dataBuf, ulInt16 \*chans, ulInt16 nChans )**

вводит 'nChans' отсчетов (не более 32) с каналов АЦП, указанных в массиве 'chans' в массив 'dataBuf'.

При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **ulInt16 ADChanMulti( int16 \*dataBuf, ulInt16 nPoints, ulInt16 chan, ulInt16 rate )**

считывание 'nPoints' отсчетов по каналу 'chan' в массив 'dataBuf' с интервалом 'rate' мкс между отсчетами (синхронизация от таймера). 'rate' = 14..65535 мкс (14 мкс соответствует 71.4КГц).

При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **ulInt16 ADChansMulti( int16 \*dataBuf, ulInt16 nCuts, ulInt16 \*chans, ulInt16 nChans, ulInt16 rate )**

считывание 'nCuts' кадров в массив 'dataBuf' с интервалом 'rate' мкс между кадрами (синхронизация от таймера). Под кадром подразумевается ряд отсчетов, состоящий из результатов последовательного опроса 'nChans' каналов, указанных в массиве 'chans'.

Минимальное значение 'rate' равно  $14 \cdot nChans$ . При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **ulInt16 SetSeqChans( ulInt16 \*chans, ulInt16 nChans )**

установка последовательности опроса 'nChans' каналов, указанных в массиве 'chans'.

При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **ulInt16 SetRate( ulInt16 rate )**

установка частоты опроса каналов, 'rate' = 14..65535 мкс (14 мкс соответствует 71.4КГц).

При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **ulInt16 ADFlow( void )**

запуск АЦП в циклическом режиме опроса каналов. Может быть использован для возобновления работы АЦП без переустановки каналов и частоты опроса.

При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **int16 ADWaitRead( waitTime : ulInt16 )**

ожидает готовности преобразования АЦП и возвращает результат. Параметр 'waitTime' задает период ожидания в мкс, по истечении этого периода (таймаут) ф-ция возвращает код 7FFFh(32767).

#### **int16 ADStop( void )**

останавливает режим работы АЦП, заданный ф-цией ADFlow.

При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

### **ФУНКЦИИ ПО РАБОТЕ С ЦИФРОВЫМИ (ДИСКРЕТНЫМИ) ТТЛ ЛИНИЯМИ**

#### **ulInt16 DigIn( void )**

возвращает состояние внешних цифровых TTL линий.

При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

#### **ulInt16 DigOut( ulInt16 dCode )**

функция установки внешних цифровых TTL линий, 'dCode' - соответствующий код.

При обнаружении таймаута ф-ция возвращает код 7FFFh (32767).

### **ФУНКЦИИ ДЛЯ РАБОТЫ С ПРЕРЫВАНИЯМИ**

#### **void IntrSetup( ulInt16 irqN )**

выбор номера прерывания IRQ, irqN должно быть равно 2, 3, 4 или 7 и соответствовать установке перемычек на плате.

#### **void InitIntr( ulInt16 rate, void \*vector )**

программирует контроллер прерываний и плату для работы в режиме генерации прерываний.

## **void StopIntr(void)**

выключает режим генерации прерываний на плате и восстанавливает контроллер прерываний PC.

## **ОПИСАНИЕ БИБЛИОТЕКИ ФУНКЦИЙ WINDOWS**

Для программистов, собирающихся писать свои собственные программы для работы с платой АЦП в среде Microsoft Windows предназначен комплект WinLib95.

Основной частью пакета является 32-разрядная DLL-библиотека w1293g.dll. Она представляет собой библиотеку функций, которая может быть использована совместно с большинством существующих систем программирования в среде Windows (в комплект поставки входят примеры ее использования в среде Borland Delphi 3.0, LabView 5.1). Библиотека подпрограмм позволяет Вам использовать практически все возможности платы, не вдаваясь в тонкости программирования на уровне ассемблера и портов ввода - вывода. В том случае, если Вы собираетесь сами программировать плату через порты ввода - вывода, то наша библиотека может быть использована Вами в качестве законченного и отлаженного примера, на основе которого Вы можете реализовать свои алгоритмы.

Библиотека содержит функции, позволяющие осуществлять ввод - вывод аналоговой и цифровой информации в асинхронном режиме, вводить и выводить аналоговую информацию как в одноканальном так и в многоканальном режимах с произвольной синхронизацией ввода. Мы надеемся, что описываемая ниже библиотека упростит и ускорит написание Ваших программ.

Библиотека поставляется в 32-битном варианте.

### **DLL-библиотеки в Microsoft Windows**

DLL-библиотека (Dynamic Link Library) представляет собой набор функций, доступных для любой программы в среде Windows. Программа может вызвать функцию из DLL, просто указав при вызове имя DLL и имя функции либо ее порядковый номер. Причем на этапе трансляции программы пользователя DLL не участвует, т.е. связывание программы пользователя и DLL происходит на этапе выполнения. Фактически, одновременно с загрузкой любой программы, все используемые DLL будут найдены на диске и загружены в память; это происходит абсолютно незаметно для пользователя. Большим преимуществом DLL является то, что для них определен стандартный способ передачи параметров, не зависящий от конкретного транслятора; фактически это означает, что функции библиотеки могут быть одинаково легко использованы в программах на Borland C++, Microsoft Visual C, Borland Delphi, LabView, Visual Basic (и практически в любой другой среде программирования под Windows).

После того, как Вы подключили библиотеку к Вашей программе, присутствие файла w1293g.dll на диске необходимо для работы Вашей программы. DLL может находиться в одной из следующих директорий:

- в директории вместе с Вашей программой (т.е. ехе-файлом);
- в директории .\WINDOWS\SYSTEM\ (настоятельно рекомендуется !!!);
- в какой-либо другой директории, указанной в переменной PATH.

В противном случае Windows при загрузке Вашей программы не сможет найти DLL, и Вы увидите соответствующее сообщение об ошибке. Во избежание проблем, связанных с последующей установкой новых версий библиотек, желательно хранить библиотеки в одном месте, например, в директории .\WINDOWS\SYSTEM\.

Конкретный способ использования DLL в системе программирования может быть различным.



## ОПИСАНИЕ БИБЛИОТЕКИ ФУНКЦИЙ LabVIEW

### Использование библиотеки wl1293g.dll в среде графического программирования LabView.

Для программирования задач ввода-вывода и обработки сигналов, не требующих высокого быстродействия обслуживающей программы, чрезвычайно удобной является система LabView фирмы National Instruments. Среда графического программирования позволяет пользователю создавать программу, "рисую" ее в специализированном редакторе как диаграмму из готовых блоков и соединяя блоки между собой. Несмотря на кажущуюся примитивность, такой способ позволяет реализовать алгоритмы достаточно высокой сложности. Так же легко создается и оформление программы - LabView предоставляет богатый выбор элементов управления, за несколько минут можно создать программу, имеющую удобный и красивый графический интерфейс. Достаточно сказать, что для того, чтобы вывести в окне график, достаточно просто мышью разместить видимый элемент "график" в нужном месте на экране, установить требуемый размер, и на диаграмме соединить вход "графика" с выходом узла, формирующего массив. И все! . Как показывает практика, создание программ в LabView - реальная задача для непрограммиста, т.к. программирование в классическом смысле знать не требуется; скорее всего, при написании программы не придется написать ни одной строчки кода. С другой стороны, для опытного программиста предоставляется возможность написать критичные участки кода на Си, Паскале или Ассемблере и включить их в программу на LabView.

Другой привлекательной чертой пакета является то, что для LabView существует большая библиотека функций, реализующих алгоритмы цифровой обработки сигналов (в том числе БПФ, цифровой фильтрация, расчет параметров сигнала, генерация различных типов сигнала). Использовать эти функции столь же просто - пиктограмма нужной функции помещается на диаграмме и соединяется с другими элементами.

Для того, чтобы в программах на LabView можно было непосредственно работать с платами АЦП-ЦАП, предоставляется несколько наборов функций (виртуальных инструментов). Эти виртуальные инструменты представляют собой соответствующим образом оформленный программный интерфейс для вызова функций из LabView. Для большинства функций написаны примеры, демонстрирующие технику использования библиотеки в LabView. Для связи LabView с DLL использовался инструмент, называемый в LabView "Call Library Function". Поставляемые в текущей версии библиотеки написаны для использования в 32-битной версии LabView 5.1.

Набор подпрограмм для работы с платами АЦП-ЦАП в среде LabView имеется в LLB комплекте adc1293.llb (LLB - формат хранения программ LabView).

Для использования предоставляемых виртуальных инструментов из библиотек LLB необходимо скопировать файл wl1293g.dll в директорию .\WINDOWS\SYSTEM.

Для получения оценочной версии и приобретения лицензионной версии LabView обращайтесь в фирму «ХОЛИТ Дэйта Системс, Лтд.»