М О Д У Л Ь *E-270*

Техническое описание
и инструкция по эксплуатации

2002 г.

19.09.02г.

ЗАО «Л-КАРД».
117105, г. Москва, Варшавское шоссе, д. 5, корп. 4, стр. 2.
lcard@lcard.ru http://www.lcard.ru
тел. (095) 785-95-25
факс (095) 785-95-14

1. Введение. ***5***

2. Назначение. ***5***

3. Состав E-270. ***5***

4. Основные параметры модуля E-270. ***6***

5. Подключение E-270. ***7***

6. Описание программного обеспечения. ***11***

6.1. Основные принципы взаимодействия с модуля E-270. **11**

6.2. Драйвера. **12**

6.3. Библиотека функций. **12**

6.3.1. Использование библиотечных функций для работы с модулем E-270. 13

6.3.2. Простой пример. 14

6.3.3. Форматы данных. 15

6.3.4. Константы и переменные. 20

6.3.5. Библиотечные функции. 21

6.4. Демонстрационные и примерные программы. **26**

6.4.6. Примерные программы. 26

6.4.7. Демонстрационные программы. 26

1. Введение.

E-270 является малогабаритным многофункциональным измерительным модулем, который может присоединяться к ПК по одному из стандартных последовательных интерфейсов.

Многообразие интерфейсов, поддерживаемых E-270, достигается применением современного многофункционального микроконтроллера CR16MCS9 фирмы National.

E-270 имеет как базовые функции (цифровой ввод/вывод, многоканальное АЦП), так и опциональные функции, позволяющие укомплектовать E-270 разными измерительными субмодулями H-27x и создать оптимальную для пользователя конфигурацию в зависимости от требуемых типов измерительных каналов (измерение тока, напряжения, сопротивления) и их количества (до 16 каналов в модуле).

E-270 создан в качестве альтернативы крупногабаритным измерительным системам на базе крейтов H-2000 и LTC.

2. Назначение.

Модуль E-270 предназначен для построения многоканальных измерительных систем сбора аналоговых данных медленно меняющихся процессов, а также цифрового управления и контроля состояния внешних устройств. Стыковка модуля с ПК осуществляется по одному из интерфейсов: RS232, RS485 или USB, что обеспечивает его совместимость с большинством современных ПК. Поставляемое с модулем ПО обеспечивает его работоспособность в среде *Windows'98/2000/XP.*

3. Состав E-270.

Таблица 1. Функциональное наполнение модуля E-270 в зависимости от его модификации.

Модификации E-270	**Интерфейсы**	**Общее количество субмодулей H-27x, устанавливаемых под заказ.**	**АЦП - 8 азрядное, 8 канальное**	**Цифровой ввод/вывод (Всего 13 линий)**	**Примечание**
E-270RS485	RS-232, RS-485	от 0 до 8	есть	есть	
E-270USB	RS-232, USB	от 0 до 8	есть	есть	
E-270CAN	RS-232, CAN	от 0 до 8	есть	есть	Перспективная модификация

Примечания:

1. В случае использования интерфейсов RS485/USB/CAN во всех модификациях E-270 интерфейс RS-232 может быть не задействован.
2. В случае, если требуется использовать только RS-232, должна использоваться модификация E-270RS485.

Таблица 2. Набор измерительных субмодулей H-27x.

Субмодуль H-27x	**Количество каналов**	**Функциональное назначение**	**Диапазон измерения**
H-27I5	2	Измеритель тока	0 … 5 мА
H-27I10	2	Измеритель тока	-10 … 10 мА
H-27I20	2	Измеритель тока	0 … 20 мА

H-27T	2	Измеритель напряжения термопары	–25 … 75 мВ
H-27R100	1	Измеритель сопротивления	0 … 100 Ом
H-27R250	1	Измеритель сопротивления	0 … 250 Ом
H-27U10	2	Измеритель напряжения	-10 … 10 В
H-27U20	2	Измеритель напряжения	0 … 20 В

Примечание: номенклатура субмодулей может пополняться.

Программное обеспечение E-270 состоит из:

- демонстрационных программ;
- драйверов для работы с модулем через различные интерфейсы;
- библиотеки функций, обеспечивающих унифицированный программный интерфейс с E-270 через различные интерфейсы.

4. Основные параметры модуля E-270.

Перед подключением E-270 следует ознакомиться с основными параметрами модуля E-270, приведенными в таблице 3.

Таблица 3. Основными параметрами модуля E-270

Параметр	**Значение**	**Примечание**
Параметры подсистемы сбора данных с субмодулей H-27x		
Количество субмодулей H-27x	до 8	Другие параметры H-27x - см. паспортные данные на соответствующий субмодуль
Максимальное количестово измерительных каналов от субмодулей H-27x с поканальной гальванической развязкой	до 16	
Частота преобразования ПНЧ в измерительных канах	250 кГц	
Программно задаваемая частота сбора данных со всех измерительных каналов	5..104 Гц	
Возможность учета калибровочных коэффициентов	Имеется	
Параметры подсистемы сбора данных с АЦП.		
Количество каналов АЦП	8	
Программно задаваемая частота сбора данных со всех каналов АЦП	5..104 Гц	
Разрядность канала данных АЦП	8	
Диапазон входного сигнала АЦП	0…(4,096±0.002)В	
Возможность учета калибровочных коэффициентов	Имеется	
Параметры подсистемы цифрового ввода-вывода.		
Количество входных линий TTL	8	
Количество выходных линий TTL	8	
Количество входных линий сигнала прерывания (TTL/CMOS)	1	
Количество дополнительных цифровых входа/выходы (выходы слаботоковые CMOS)	4	Внимание ! Допустимый выходной ток – не более 1 мА

Скорость ввода-вывода	500 байт/с	
Параметры подсистемы интерфейса с ПК.		
Максимальная скорость: - RS-232 - RS-485 - USB	 115200 бит/с; 115200 бит/с; 12 Мбит/с	 USB v1.1
Наличие гальваноразвязки интерфейсов: - RS-232 - RS-485 - USB	 НЕТ; ЕСТЬ; НЕТ	 USB v1.1
Питание E-270		
Напряжение питания: - E-270RS485 - E-270USB	 7-20 В (от внешнего источника питания) 5В (от USB)	 Есть гальваноразвязка. Нет гальваноразвязки.
Максимальная потребляемая мощность: - E-270USB - E-270RS485	 2,25 Вт 2,5 Вт	 (Ток до 450 мА от USB)
Предельно допустимые параметры		
Предельно допустимое значение напряжений на цифровых линиях ввода-вывода и входах АЦП.	**- 0.2 …+5,2 В**.	Внимание! Превышение в ходе эксплуатации предельно допустимых значений токов и напряжений влечет за собой ухудшение параметров или выход модуля из строя.
Предельно допустимые значения токов на слаботоковой CMOS линиях, настроенных на выход .	**±1 мА**	
Предельно допустимые значения тока на выходной TTL- линии ввода-вывода.	± **5 мА**	
Предельно допустимый суммарный ток нагрузки на всех цифровых выходных линиях: - при комплектации E-270, близкой к максимальной (установлено более 6 шт. субмодулей H-27x) - при неполной комплектации (6 шт. или менее H-27x)	 ±10 мА ±40 мА	
Предельно допустимые входные напряжения и токи на измерительных входах субмодулей H-27x (разъем “Аналог. вх”)	см. Паспорт соответствующего субмодуля H-27x	
Массо-габаритные характеристики		
Габарит корпуса E-270 (без подключенных кабельных разъемов)	138 x 116 x 32 мм	

5. Подключение E-270.

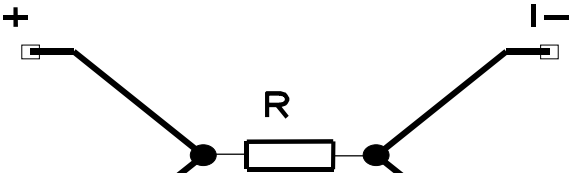
На корпусе E-270 установлено 4 разъема для внешних кабельных подключений. Среди них разъем “RS-485/USB/CAN” - интерфейсный, соответствующий выбранной модификации E-270, аналог, а на разъем “Аналог.вх” непосредственно выведены измерительные входы субмодулей H-27x, установленных внутри E-270.

Таблица 4. Описание разъемов E-270 и назначение контактов.

Интерфейс с ПК				
Разъем	Тип	Контакт	Цепь	Назначение разъема.
“RS-232”	DRB-9M	1	DSD	Интерфейс с ПК. Соответствует RS-232.
		2	RxD	
		3	TxD	
		4,7,9	не подключена	
		5	GND	
		6	DSR	
		8	CTS	
“RS-485/ USB/CAN ”	DRB-9F (Для RS-485)	1, 2	GND485	“Общий” для RS-485. Может быть использован для соединения с корпусом ПК.
		6,7	B485, A485	Интерфейс RS-485
		5	0В	0В внешнего источника питания
		4	+U	+7…20В внешнего источника питания
		3,8,9	Не подключена	
“RS-485/ USB/CAN ”	DUSB-BRA42-T11 (Для USB)	Стандартный разъем устройства USB		

Аналоговые измерительные линии субмодулей H-27x				
Разъем	*Тип*	*Номер контакта*	*Наименование сигнала в случае применения всех субмодулей H-27x, кроме H-27R*	*Наименование сигнала в случае применения субмодулей H-27R (см. примеч. снизу таблицы)*
“Аналог. Вх”	DRB-37M	1	Не подключен	Не подключен
		2	Не подключен	Не подключен
		3	Подключение электростатического экрана входных цепей (рекомендуется для повышения помехозащищенности).	
		4	Вход “-” канала 16	Вход “U-” канала 8
		5	Вход “-” канала 15	Выход “I+” канала 8
		6	Вход “-” канала 14	Вход “U-” канала 7
		7	Вход “-” канала 13	Выход “I+” канала 7
		8	Вход “-” канала 12	Вход “U-” канала 6
		9	Вход “-” канала 11	Выход “I+” канала 6
		10	Вход “-” канала 10	Вход “U-” канала 5
		11	Вход “-” канала 9	Выход “I+” канала 5
		12	Вход “-” канала 8	Вход “U-” канала 4
		13	Вход “-” канала 7	Выход “I+” канала 4
		14	Вход “-” канала 6	Вход “U-” канала 3
		15	Вход “-” канала 5	Выход “I+” канала 3
		16	Вход “-” канала 4	Вход “U-” канала 2
		17	Вход “-” канала 3	Выход “I+” канала 2
		18	Вход “-” канала 2	Вход “U-” канала 1
		19	Вход “-” канала 1	Выход “I+” канала 1
		20	Не подключен	Не подключен
		21	Не подключен	Не подключен
		22	Вход “+” канала 16	Вход “U+” канала 8
		23	Вход “+” канала 15	Выход “I-” канала 8
		24	Вход “+” канала 14	Вход “U+” канала 7
		25	Вход “+” канала 13	Выход “I-” канала 7
		26	Вход “+” канала 12	Вход “U+” канала 6
		27	Вход “+” канала 11	Выход “I-” канала 6
		28	Вход “+” канала 10	Вход “U+” канала 5
		29	Вход “+” канала 9	Выход “I-” канала 5
		30	Вход “+” канала 8	Вход “U+” канала 4
		31	Вход “+” канала 7	Выход “I-” канала 4
		32	Вход “+” канала 6	Вход “U+” канала 3
		33	Вход “+” канала 5	Выход “I-” канала 3
		34	Вход “+” канала 4	Вход “U+” канала 2
		35	Вход “+” канала 3	Выход “I-” канала 2
		36	Вход “+” канала 2	Вход “U+” канала 1
		37	Вход “+” канала 1	Выход “I-” канала 1

Примечание: В случае применения субмодуля H-27R выходы источника тока I+/- субмодуля и дифференциальный измерительные входы напряжения U+/- для соответствующего канала субмодуля должны быть соединены на измеряемом резисторе R согласно рисунку:



Входы АЦП и цифровые линии ввода-вывода				
Разъем	*Тип*	*Номер контакта*	*Назначение*	*Примечание*
“Цифр. Вх.Вых”	DRB-37F	1	Вход 1 АЦП	TTL – линии
		2	Вход 3 АЦП	
		3	Вход 5 АЦП	
		4	Вход 7 АЦП	
		5	Аналоговая земля АЦП	
		6	Внешняя линия прерывания	
		7	Цифр. выход 1	TTL – линии
		8	Цифр. выход 3	
		9	Цифр. выход 5	
		10	Цифр. выход 7	
		11	Цифр. вход 1	
		12	Цифр. вход 3	
		13	Цифр. вход 5	
		14	Цифр. вход 7	
		15	Цифр. вход/выход 8	Слаботоковые линии CMOS
		16	Цифр. вход/выход 10	
		17, 18, 19	Цифровая земля	
		20	Вход 0 АЦП	TTL – линии
		21	Вход 2 АЦП	
		22	Вход 4 АЦП	
		23	Вход 8 АЦП	
		24	Аналоговая земля АЦП	
		25	Не подключен	
		26	Цифр. выход 2	TTL – линии
		27	Цифр. выход 4	
		28	Цифр. выход 6	
		29	Цифр. выход 8	
		30	Цифр. вход 2	
		31	Цифр. вход 4	
		32	Цифр. вход 6	

		33	Цифр. вход 8	
		34	Цифр. вход/выход 9	Слаботоковые линии CMOS
		35	Цифр. вход/выход 11	
		36, 37	Цифровая земля	

Примечание: Контакты аналогой и цифровой земли гальванически связаны внутри E-270.

6. Описание программного обеспечения.

6.1. Основные принципы взаимодействия с модуля E-270.

Обмен с модулем осуществляется по принципу запрос-ответ. Каждый модуль имеет свой логический адрес, присваиваемый ему на стадии конфигурирования. Компьютер посылает запрос модулю, содержащий: логический адрес интересующего модуля, код команды и данные, необходимые для выполнения команды. Приняв запрос, модуль посылает ответ только в том случае, если запрос принят без ошибок (контролируется по контрольной сумме), код команды соответствует одной из поддерживаемых команд, адрес в запросе соответствует адресу модуля или равен нулю. Во всех остальных случаях запрос игнорируется, и модуль ожидает прихода следующего запроса. Нулевой адрес является универсальным и отвечать на него обязан любой модуль (обработка запросов и схема адресации модулей осуществляется независимо от типа интерфейса, поэтому не следует путать адрес модуля и адрес USB устройства и в частности нулевой адрес модуля и нулевой адрес USB устройства). Время обработки запроса <=200 мсек.

Запросы, посылаемые компьютером, подразделяются на две группы. Первая - состоит из запросов конфигурирования модуля:

- получить/установить конфигурацию модуля
- сохранить текущую конфигурацию модуля в EEPROM
- разрешить запись в EEPROM модуля/мезонина
- чтение/запись EEPROM модуля/мезонина

Вторая – объединяет в себе запросы для работы с цифровыми и аналоговыми каналами модуля:

- получить/установить состояние цифровых линий ввода/вывода
- чтение значения счетчика внешних событий
- чтение данных от АЦП

Основным состоянием модуля является сбор аналоговых данных. При получении запросов первой группы модуль временно приостанавливает сбор данных и обрабатывает запрос, после чего возвращается в основное состояние. Запросы второй группы обрабатываются без приостановки сбора. Данная особенность обработки команд позволяет исключить команды на запуск и останов АЦП, что упрощает работу с модулем.

При сборе аналоговых данных в ОЗУ модуля создается буфер, накапливающий «блоки данных». Каждый такой «блок данных» занимает 47 байт и состоит из: данных 16 каналов субмодулей, данных 8 каналов 8-ми битного АЦП, данных о состояния линий цифрового ввода/вывода и времени окончания аналого-цифрового преобразования (см. ниже структуру DATA). Размер буфера (в байтах) определяется частотой дискретизации по следующей формуле:

$$размер\ \ буфера\ \ в\ \ байтах\ \ = 47 * 2 * \left[\frac{частота\ \ дискретизации}{5} \right]_{\text{int}}$$

Таким образом, в буфере могут находится данные за последние ~ 400 мсек. Для обмена с компьютером используется стандартная схема. Буфер делится пополам, первая половина используется для сбора данных, в то время как вторая половина используется для обмена с компьютером. По мере заполнения первой половины «блоками данных» половинки буфера меняются своими функциями. Если данные из второй половины буфера не были считаны до этого момента, то они будут безвозвратно утеряны. Поэтому надо обеспечить формирование запросов на чтение данных не реже чем раз в 200 мсек. Запросы могут приходить и чаще, в таком случае первый запрос будет обслужен сразу, а второй будет ожидать окончания заполнения первой половины буфера. В одном запросе компьютеру передаются $\left[\frac{частота\ \ дискретизации}{5} \right]_{\text{int}}$ блоков данных.

При включении питания модуль считывает начальную конфигурацию, записанную в EEPROM модуля (адрес модуля, частоту дискретизации и флаги использования коррекции) и при непротиворечивости считанных данных и правильной контрольной суммы переходит в режим сбора аналоговых данных в соответствии с выбранными параметрами. При ошибочной контрольной сумме выбираются параметры по умолчанию: адрес модуля 0, частота дискретизации 5 Гц, коррекция отключена.

6.2. Драйвера.

Библиотека функций общается с модулем через тот или иной драйвер в зависимости от выбранного интерфейса подключения модуля. Для связи с модулем через RS232 или RS485 интерфейс используется стандартный драйвер.

Для связи через USB-интерфейс следует установить драйвер, находящийся в каталоге \E270\USBDRV. Данный драйвер предназначен для ОС *Windows'98/2000.* Установка осуществляется обычным способом:

1. устройство подключается к компьютеру. (так как устройства с USB-интерфейсом обеспечивают горячее подключение/отключение, то подключать его можно в любой момент)
2. после того как ОС опознает устройство, укажите местоположения файла содержащего информацию об устройстве –> \E270\USBDRV*.inf
3. произведите перезагрузку, если требуется и удостоверьтесь что в разделе «свойства системы\контроллеры шины USB» появилось устройство E270.

6.3. Библиотека функций.

Библиотека функций обеспечивает унифицированный программный интерфейс для работы с модулем через различные физические интерфейсы. Исходный текст библиотеки написан на языке ANSI-C и находится в каталоге \E270\DLL. В этом же каталоге находится заголовочный файл e270api.h и библиотечный файл e270api.dll. Каталог \E270\DLL\BORLAND\ содержит файл библиотеки импорта e270api.lib для использования в среде Borland Builder\Borland C++. Каталог \E270\DLL\MSVC\ содержит файл библиотеки импорта e270api.lib для использования в среде Microsoft Visual C++. Каталог \E270\DLL\ DELPHI\ содержит файл e270api.pas для использования в среде Borland Delphi.

6.3.1. Использование библиотечных функций для работы с модулем E-270.

Для какого-либо взаимодействия с модулем посредством библиотечных функций необходимо выполнить следующие действия:

- поместить библиотечный файл e270api.dll в один из следующих каталогов: текущий, windows\system, каталог указанный в переменной окружения path.
- подключить библиотеку к проекту, воспользовавшись файлами e270api.lib и e270api.h или e270api.pas
- создать экземпляр структуры E270 и проинициализировать его, вызвав функцию E270Init()
- задать тип интерфейса в поле E270.Interface (для связи через RS232/RS485 следует так же задать номер com-порта в поле E270.Port) и адрес интересующего вас модуля в поле E270.address
- установить соединение с модулем, произведя вызов функции E270Open()
- осуществить требуемые действия (см. ниже)
- разорвать соединение с модулем, вызвав функцию E270Close()

Для сбора аналоговых данных, после установления соединения с модулем, необходимо выполнить следующие действия:

- задать параметры сбора данных: делитель частоты дискретизации в поле E270.fdivisor и разрешить/запретить использование корректирующих коэффициентов, установив/сбросив соответствующие флаги в поле E270.flags
- сконфигурировать модуль в соответствии с выбранными параметрами вызвав функцию E270SetConfig()
- периодически (не реже чем раз в 200 мс) забирать из модуля данные, вызывая функцию E270GetData().

Для управления цифровыми линиями ввода/вывода, после установления соединения с модулем, необходимо выполнить следующие действия:

- установит/считать состояние линий ввода/вывода вызвав функцию E270SetDIO()/E270GetDIO()

Для работы со счетчиком внешних событий, после установления соединения с модулем, необходимо выполнить следующие действия:

- задать параметры счетчика внешних событий: срабатывание по переднему/заднему фронту импульса и разрешение/запрет синхронизации сбора аналоговых данных, при обнаружении соответствующего события, установив/сбросив соответствующие флаги в поле E270.flags
- сконфигурировать счетчик внешних событий, вызвав функцию E270SetConfig()
- считывать значение счетчика, вызывая функцию E270GetGMSC_ECNT().

Для чтения содержимого EEPROM модуля/мезонина после установления соединения с модулем необходимо выполнить следующие действия:

- вызвать функцию E270GetEEPROM()
- проверить достоверность содержимого EEPROM вызвав функцию E270CRCCheck()

Для записи содержимого EEPROM модуля/мезонина после установления соединения с модулем необходимо выполнить следующие действия:

- изменить содержание поля E270.eeprom/E270.mezzanine[…].eeprom в соответствии с требуемым результатом

- разрешить запись в EEPROM вызвав функцию E270WriteEnableEEPROM()
- записать EEPROM вызвав функцию E270SetEEPROM()

Для сохранения текущей конфигурации модуля, после установления соединения с модулем, необходимо выполнить следующие действия:

- вызвать функцию E270SaveConfig()

6.3.2. Простой пример.

```
//*******************************************************
// В данном примере осуществляется сбор аналоговых данных,
//         используя функции библиотеки E270API
//*******************************************************
#include <malloc.h>                                    //
#include <stdio.h>                                     //
#include "..\dll\e270api.h"                             // файл содержащий заголовки
                                                        // библиотечных функций
// входные параметры
#define ACQ_TIME   15.3                                 // время в течении которого
                                                        // осуществляется сбор данных (сек)
#define FREQ       100                                  // частота дискретизации (Гц)
#define CORRECTION 0                                    // корректировка данных ацп
#define FORMAT     1                                   // формат отображения данных
                                                       //  0 - в кодах ацп
                                                       //  1 - в физических величинах
#define ADDRESS    0                                    // адрес модуля
#define INTER      INTERFACE_USB                        // тип интерфейса
#define PORT       2                                    // номер COM-порта при связи
                                                        // через RS232|RS485
// производные величины
#define N          (WORD)(ACQ_TIME*FREQ)                // общее число блоков данных ацп/цвв
                                                        // для приема
#define FDIVISOR   (WORD)(PNCHFREQ/FREQ)                // делитель частоты дискретизации
#define DN         (WORD)(FREQ/5)                       // число блоков данных ацп/цвв
                                                       // передаваемых в одном запросе
//*******************************************************
//
//*******************************************************
int main(int argc, char* argv[])
{
 WORD i, j, n=0;                                        // вспомогательные параметры
 E270 module;                                           // основная конфигурационная структура
 DATA data[DN];                                         // буфер данных ацп
    // инициализация полей структуры
    E270Init(&module);
    // задаем адрес модуля
    module.address=ADDRESS;
    // выбираем тип интерфейса
    module.Interface=INTER;
     // задаем номер COM-порта
    module.Port=PORT;
    // пытаемся связаться с модулем
    if(!E270Open(&module)) {
        // считываем и проверяем на достоверность EEPROM модуля
        // отображаем название и серийный номер модуля
        if(!E270ReadEEPROM(&module, 0) && !E270CRCCheck(&module, 0))
          printf("## Модуль: %.7s s\\n: %.9s\n", module.eeprom.name, module.eeprom.sn);
        else printf("## Модуль: XXXXXXX s\\n: XXXXXXXXX\n");
        // отображаем названия установленных мезонинов
        for(i=0; i<8; i++) {
          for(j=0; j<10 && IDString[j].id!=module.mezzanine_id[i]; j++);
          printf(">> место %i модуль %4s (%s)\n", i+1, IDString[j].name, IDString[j].unit);
        }
        // задаем делитель частоты дискретизации
       module.fdivisor=FDIVISOR;
       // задаем флаги использования коррекции данных ацп
       module.flags=CORRECTION_ZERO|CORRECTION_COEF;
       // устанавливаем конфигурацию модуля в
```

```
        // соответствии с выбранными параметрами
        if(!E270SetConfig(&module, 0)) {
            printf(">> Начат сбор данных\n");
            // небольшая задержка необходимая для
            // начальной буферизации данных внутри модуля
            Sleep(100);
            //
            while(n<N)
                // запрашиваем данные от АЦП
                if(E270GetDATA(&module, data)) break;
                else
                    // отображаем результат ацп
                    // (в одном запросе передается DN
                    // блоков данных ацп/цвв)
                    for(i=0; n<N && i<DN; n++, i++) {
                        // отображаем временную метку соответствующую завершению ацп
                        printf("%.10i>", data[i].gmsc);
                        // отображаем данные 14-ти и 8-ми битного ацп
                        for(j=0; j<24; j++) {
                        //*** вывод результатов ацп в кодах ацп
                        if(!FORMAT)
                          if(j<16) printf(" %04X", (int)E270GetValue(&module, data+i, j, 0));
                              else printf(" %02X", (int)E270GetValue(&module, data+i, j, 0));
                        //*** вывод результатов ацп в физических величинах
                        else
                          if(j<16) printf(" %6.2f", E270GetValue(&module, data+i, j, 1));
                             else printf(" %4.2f", E270GetValue(&module, data+i, j, 1));
                        // выводим данные цифровых линий вода/вывода
                        printf("  %.2X %.2X %.2X\n",
                               (int)data[i].dout, (int)data[i].din, (int)data[i].dio);
                    }
            // сбор данных остановлен
            if(n==N) printf(">> Сбор данных успешно завершен\n");
                else printf(">> Ошибка связи с модулем. Сбор данных прерван.\n");
        } else printf(">> Ошибка конфигурирования модуля\n");
        // закрываем интерфейс
        E270Close(&module);
    } else printf(">>> Ошибка открытия интерфейса\n");
 return 0;
}
```

6.3.3. Форматы данных.

6.3.3.1. Основная конфигурационная структура.

Структура E270 – основная структура, содержащая всю необходимую информацию о конфигурации модуля и состоянии связи по выбранному интерфейсу. Эта структура используется во всех библиотечных функциях обмена с модулем.

Поля конфигурационной структуры можно логически разделить на три группы:

- ***состояние интерфейса***

  поля данной группы отвечают за выбор типа используемого для связи с модулем интерфейса, показывают текущее состояние выбранного интерфейса и содержат статистику обработки запросов. (Примечание: не следует изменять значение полей данной группы при открытом интерфейсе)

- ***конфигурация модуля***

  поля данной группы содержат конфигурацию модуля, которая была считана из самого модуля или ожидает записи в модуль.

  Адрес модуля – логический адрес модуля, с которым установлена связь (см. E270SetConfig)

  Делитель частоты дискретизации может принимать значения 240..5000 и вычисляется по следующей формуле:

$$делитель = \left[\frac{25000}{частота \quad дискретизации} \right]_{\text{int}}$$

Флаги конфигурации модуля позволяют: сбросить внутренний таймер модуля и/или счетчик внешних событий, настроить счетчик внешних событий на срабатывание по переднему/заднему фронту импульса, разрешить использование корректировочных коэффициентов, считанных из EEPROM субмодулей (см. таблицу 5).

Корректировка данных субмодулей выполняется по следующей формуле:

$$S' = \frac{S * Sc}{32768} + Zc \text{, где}$$

S' - откорректированное значение

S - исходное значение, приведенное к диапазону 0..32767

Sc - масштабный коэффициент (считывается из EEPROM мезонина или равен 32768, если корректировка масштаба запрещена)

Zc - смещение нуля (считывается из EEPROM мезонина или равен 0, если корректировка нуля запрещена)

- ***содержимое EEPROM модуля и мезонинов***

  поля данной группы содержат данные, считанные из EEPROM модуля/мезонина или данные ожидающие записи в EEPROM модуля/мезонина

```
typedef struct{                    //
    //**** состояние интерфейса.
    //**** примечание: не следует изменять данные настройки при открытом интерфейсе.
    BYTE   Interface;          // тип используемого интерфейса:  RS232, RS485, USB
    BYTE   Port;             // номер COM порта для интерфейса RS232 или RS485
    HANDLE Handle;       // указатель на открытый интерфейс
    DWORD  ReqTotal;           // общее число переданных через интерфейс запросов
    DWORD  ReqBad;                    //  число запросов, на которые не пришло подтверждения
    //**** конфигурация модуля
    //**** примечание: после изменения конфигурации следует вызывать функцию
    //**** E270SetConfig()  чтобы внесенные изменения возымели действие.
    BYTE status;               // флаги состояния модуля (не используются)
    BYTE address;                  // адрес модуля
    BYTE flags;                        // флаги конфигурации модуля (см. таблицу 5)
    WORD fdivisor;                 // делитель частоты дискретизации 240..5000 (104..5 Гц)
                                   // частота дискретизации = PNCHFREQ /fdivisor

    //**** состояние счетчика внешних событий и счетчика миллесекунд
    //**** примечание: данные поля обновляются при вызове функции E270GetGMSC_ECNT()
    DWORD  gmsc;                       //  время (в мсек), прошедшее с момента включения модуля
    DWORD  ecnt;                     // счетчик внешних событий
    BYTE mezzanine_id[8];       // идентификаторы установленных мезонинов
    //**** содержимое EEPROM модуля
    //**** примечание: после изменений данного поля следует вызывать функции
```

```
//**** E270WriteEnableEEPROM() и E270WriteEEPROM() чтобы внесенные изменения
//**** возымели действие.
 struct eeprom_e270{
      //*** системная область.
      //*** примечание: запись в эту область заблокирована
      WORD  size;                  // размер системной области
      BYTE  sn[9];                 // серийный номер модуля
      BYTE  name[7];               // название изделия
      BYTE  revision;          // ревизия     изделия
      BYTE  cpu_type[5];               // тип установленного CPU/DSP
      DWORD qfrequency;          // частота системного кварца в Гц
      BYTE  reserved2[3];          // зарезервировано
      BYTE  crc_s;                 // контрольная сумма
      //*** начальная конфигурация модуля
      //*** примечание: содержимое EEPROM по данным адресам обновляется при
      //*** вызове функций E270SaveConfig() и E270WriteEEPROM
      BYTE address;                    // адрес модуля
      WORD fdivisor;             // частота дискретизации
        BYTE flags;                    // флаги конфигурации модуля (см. таблицу 5)
      BYTE dout;               // начальное состояние линий вывода
      BYTE   dio;                              // начальная конфигурация и состояние дополнительных линий цифрового
                              // КМОП ввода/вывода
      BYTE crc_i;                // контрольная сумма
      //*** свободная область
      //*** примечание:  данная  область  доступна  для  хранения пользовательской
      //*** информации.
      BYTE free[88];                   // свободно
      BYTE crc_f;                // контрольная сумма
       } eeprom;                 //
struct mezzanine_t{                    //
       //*** содержимое EEPROM мезонинов
      //*** примечание: после изменений данного поля следует вызывать функции
      //*** E270WriteEnableEEPROM() и E270WriteEEPROM() чтобы внесенные
      //*** изменения  возымели действие.
       struct eeprom_t {                 // eeprom мезонина                    |
            //*** системная область.
            //*** примечание: запись в эту область заблокирована
            WORD ver;              // версия флеша
            WORD id;           // идентификатор мезонина
            WORD crc;              // контрольная сумма
            BYTE serial[6];        // серийный номер    мезонина
            WORD koef[4];          // поправочные коэффициенты
            //*** свободная область
            //***  примечание:  данная  область  доступна  для  хранения пользовательской
            //*** информации.
            WORD free[54];           // свободно
       } eeprom;                     //                                    |
```

```
  } mezzanine[8];                    // номер посадочного места 0..7
 } E270, *PE270;                  //
```

Таблица 5. Байт кофигурации модуля.

Номер бита	Назначение
0	0 – отключить коррекцию смещения нуля данных субмодулей. 1 – включить коррекцию смещения нуля данных субмодулей. коррекция показаний каждого конкретного модуля оссуществляется при следующих условиях: контрольная сумма EEPROM субмодуля соответствует содержимому EEPROM или установлен бит принудительного использования коррекции (бит 2)
1	0 – отключить коррекцию масштаба данных субмодулей. 1 – включить коррекцию масштаба данных субмодулей. коррекция показаний каждого конкретного модуля оссуществляется при следующих условиях: контрольная сумма EEPROM субмодуля соответствует содержимому EEPROM или установлен бит принудительного использования коррекции (бит 2)
2	0 – не использовать принудительную коррекцию данных субмодулей. (использование коррекции зависит от состояния битов 0, 1 и достоверности содержимого EEPROM, соответствующего субмодуля) 1 – использовать принудительную коррекцию данных субмодулей. (использование коррекции зависит только от состояния битов 0 и 1)
3	Зарезервированно
4	0 – выбрать в качестве внешнего события переход 0 -> 1 1 - выбрать в качестве внешнего события переход 1 -> 0
5	0 – синхронизация сбора аналоговых по внешнему событию отключена 1 – включить синхронизацию сбора аналоговых данных по выбраному внешнему событию. При обнаружении внешнего события будет оссуществлятся перезапуск процесса аналого-цифрового преобразования.
6	1 – сбросить внутренний таймер модуля. Данный бит автоматически устанавливается в 0 после сброса таймера.
7	1 – сбросить счетчик внешних событий. Данный бит автоматически устанавливается в 0 после сброса счетчика.

6.3.3.2. Структура «блока данных».

Как было отмечено ранее ***(п. 3.1)***, при сборе аналоговых данных в ОЗУ модуля накапливаются результаты аналого-цифрового преобразования в виде «блоков данных». «Блок данных» состоит из:

- ***данных субмодулей***
  для хранения данных субмодулей отводятся 16 16-ти битных беззнаковых слова, причем нескорректированные значения данных субмодулей принимают значения из диапозона 0..32767 (независимо от частоты дискретизации), скорректированные значения, из-за наличия смещения нуля и массштабного коэффициента, могут принимать значения из диапазона 0..65535.

  Внимание: *при самостоятельном анализе данных субмодулей следует учесть следующую особенность представления данных: значения 0..49151 итерпритируются как беззнаковые целые, значения 49152..65535 итерпритируются как знаковые целые в дополнительном коде. Библиотечная функция E270GetValue() учитывает данную особеность представления данных.*

- ***данных 8-ми битного АЦП***
  отсчеты 8-ми битного АЦП принимают значение 0..255 и для их хранения выделяется массив из 8 беззнаковых байт.
- ***данных состояния линий цифрового ввода/вывода***
  один байт отводится для хранения состояния линий цифрового ТТЛ вывода, один байт для хранения состояния линий цифрового ТТЛ ввода и один для хранения конфигурации и состояния линий дополнительного цифрового КМОП ввода/вывода.
- ***времени окончания аналого-цифрового преобразования***
  после окончания процесса аналого-цифрового преобразования к блоку данных ацп/цвв добавляется значение внутреннего счетчика времени, отсчитывающего время (в миллисекундах) прошедшее с начала включения модуля. Под данное значение отводится 32-х разрядное беззнаковое слово.

Структура DATA используется для хранения данных полученных при ацп.

```
typedef struct DATA_T          //  «блок данных»
{                              //
      DWORD   gmsc;            // число миллисекунд прошедшее с момента включения
     модуля до
                               // окончания процесса цифро-аналогого преобразования
      short    pnch[16];   //  данные субмодулей
      BYTE   adc8[8];         //  данные 8-ми битного АЦП
      BYTE   dout;            // состояние линий цифрового ТТЛ ввода
      BYTE   din;       // состояния линий цифрового ТТЛ вывода
      BYTE    dio;                    // конфигурация и состояние дополнительных линий
     цифрового
                               // КМОП ввода/вывода (см. таблицу  6)
} DATA, *PDATA;        //
```

Таблица 6. Байт кофигурации и состояния дополнительных линий цифрового КМОП ввода/вывода.

Номер бита	Назначение
0	Конфигурация линии 1 дополнительного цифрового КМОП ввода/вывода 0 – линия работает как вход, 1 – линия работает как выход
1	Конфигурация линии 2 дополнительного цифрового КМОП ввода/вывода 0 – линия работает как вход, 1 – линия работает как выход
2	Конфигурация линии 3 дополнительного цифрового КМОП ввода/вывода 0 – линия работает как вход, 1 – линия работает как выход
3	Конфигурация линии 4 дополнительного цифрового КМОП ввода/вывода 0 – линия работает как вход, 1 – линия работает как выход
4	Состояние линии 1 дополнительного цифрового КМОП ввода/вывода
5	Состояние линии 2 дополнительного цифрового КМОП ввода/вывода
6	Состояние линии 3 дополнительного цифрового КМОП ввода/вывода
7	Состояние линии 4 дополнительного цифрового КМОП ввода/вывода

6.3.3.3. Идентификационная структура.

Данная структура позволяет по идентификационному коду мезонина получить его название и физические единицы измерения в виде текстовой строки.

Структура IDSTRING используется для определения массива идентификаторов мезонинов

```
typedef struct{                  //
    BYTE id;                    // идентификатор мезонина (одна из констант H27__XXX)
    char name[5];                // идентификатор мезонина в текстовом виде
    char unit[5];                // единицы измерения физической величины
} IDSTRING;                      //
```

6.3.4. Константы и переменные.

```
// константа для пересчета частоты дискретизации
#define PNCHFREQ     25000
// константы задающие тип интерфейса
#define INTERFACE_RS232 0
#define INTERFACE_RS485 1
#define INTERFACE_USB   2
#define INTERFACE_CAN   3
// размер EEPROM модуля/мезонина в байтах
#define EEPROM_SIZE     128
// идентификаторы мезонинов
#define H27__U10            10
#define H27__U20            11
#define H27__U30            12
#define H27__U300           13
#define H27__I5              20
#define H27__I10             21
#define H27__I20             22
#define H27__R               30
#define H27__T               40
#define H27__EMPTY    255
// флаги достоверности данных в различных областях EEPROM модуля
// комбинация данных флагов возвращается функцией E270CRCCheck()
#define EEPROM_CRC_S   1
#define EEPROM_CRC_I    2
#define EEPROM_CRC_F   4
// флаги конфигурации модуля
// комбинация данных конфигурационных флагов хранится/задается в поле flags (см. таблицу 5)
#define CORRECTION_ZERO                 0x01      // учитывать коэффициент коррекции нуля внутри модуля
#define CORRECTION_COEF                 0x02      // учитывать коэффициент коорекции масштаба  внутри модуля
#define CORRECTION_ALWAYS               0x04      // использовать принудительную коррекцию
#define EVENT_EDGE                      0x10      // срабатывание счетчика внешних событий :
                                                  // 0 (0->1) или 1 (1->0)
#define EVENT_SYNCHRONIZATION     0x20            // разрешить использовать внешнее событие для синхронизации
                                                  // сбора аналоговых данных
```

```
#define RESET_GMSC                      0x40    // сбросить внутренний таймер
#define RESET_ECNT                      0x80    // сбросить счетчик внешних событий
// массив  содержащий идентификаторы мезонинов и единицы
// измерения физических величин в текстовом виде
extern const IDSTRING IDString[10];
```

6.3.5. Библиотечные функции.

Примечание: если не оговорено явно, то при удачном завершении функции возвращают 0 иначе –1 и формируется текстовое сообщение об ошибке в строке E270APIMessage.

6.3.5.1. Функции инициализации и открытия/закрытия интерфейса.

Формат: ***int E270Init(PE270)***
Назначение: Начальная инициализация полей структуры. Данную функцию следует вызывать сразу после создания экземпляра структуры E270 и до вызова каких либо других функций.
Передаваемые параметры: • указатель на инициализируемую структуру

Формат: ***int E270Open(PE270)***
Назначение: Установка соединения с выбранным модулем. Перед вызовом данной функции следует проинициализировать поля Interface, Port, address.
Передаваемые параметры: • указатель на структуру конфигурации модуля.

Формат: ***int E270Close(PE270)***
Назначение: Прервать соединение с модулем.
Передаваемые параметры: • указатель на структуру конфигурации модуля.

6.3.5.2. Функции конфигурирования модуля.

Формат: ***int E270SaveConfig(PE270)***
Назначение: Сохранить текущую конфигурацию модуля. При вызове данной функции происходит сохранение флагов конфигурации модуля, делителя частоты дискретизации, состояние линий цифрового ввода/вывода и адреса модуля в EEPROM. При включении питания модуля данные параметры восстанавливаются из EEPROM.
Передаваемые параметры: • указатель на структуру конфигурации модуля.

Формат: ***int E270GetConfig(PE270)***
Назначение: Получить текущую конфигурацию модуля. При успешном завершении данной функции обновляются следующие поля конфигурационной структуры: делитель частоты дискретизации, флаги конфигурации модуля, адрес модуля. Примечание: Так как модуль обрабатывает только запросы содержащие в заголовке свой или нулевой адрес, то поле address изменят свое значение только когда происходит запрос по нулевому адресу.
Передаваемые параметры: • указатель на структуру конфигурации модуля.

Формат: ***int E270SetConfig(PE270, BYTE)***
Назначение: Установить конфигурацию модуля. При успешном завершении данной функции модуль меняет кофигурацию счетчика внешних событий, частоту дискретизации и начнет использовать коррекцию данных в соответствии с параметрами заданными полями fdivisor и flags. Кроме того, если второй входной параметр отличен от нуля, то происходит смена адреса модуля и значения поля address конфигурационной структуры.
Передаваемые параметры: • указатель на структуру конфигурации модуля. • новый адрес модуля (если этот параметр равен нулю, то адрес модуля не меняется)

6.3.5.3. Функции чтения данных от АЦП.

Формат: ***int E270GetDATA(PE270, PDATA)***
Назначение: Чтение данных от АЦП. При успешном завершении данной функции в приемном буфере будут находиться (int)(PNCHFREQ/5/fdivisor) «блоков данных», поэтому размер буфера должен быль не менее sizeof(DATA)*(int)(PNCHFREQ/5/fdivisor).
Передаваемые параметры: • указатель на структуру конфигурации модуля. • указатель на приемный буфер.

Формат: ***int E270GetDIO(PE270, BYTE*, BYTE*, BYTE*)***

Назначение: Получить состояние цифровых линий ввода/вывода. При успешном завершении данной функции второй входной параметр будет содержать значение соответствующее состоянию линий цифрового ТТЛ вывода, третий – значение соответствующее состоянию линий цифрового ТТЛ ввода, четвертый – значение соответствующее конфигурации и состоянию линий дополнительного цифрового КМОП ввода/вывода.

Передаваемые параметры:

- указатель на структуру конфигурации модуля.
- указатель на байт в котором будет сохранено состояние линий цифрового ТТЛ выхода.
- указатель на байт в котором будет сохранено состояние линий цифрового ТТЛ входа.
- указатель на байт в котором будет сохранена конфигурация и состояние линий вспомогательного цифрового КМОП ввода/вывода.

Формат: ***int E270SetDIO(PE270, BYTE, BYTE)***

Назначение: Изменить кофигурацию и состояние линий цифрового ввода/вывода. При успешном завершении данной функции на линиях цифрового ТТЛ вывода будет выставлено значение переданное в качестве второго параметра, а линии вспомогательного цифрового КМОП ввода/вывода будут сконфигурированны в соответствии со значением третьего параметра (см. таблицу 6).

Передаваемые параметры:

- указатель на структуру конфигурации модуля.
- новое состояние линий цифрового ТТЛ вывода.
- новая конфигурация и состояние линий вспомогательного цифрового КМОП ввода/вывода.

6.3.5.4. Чтение значения счетчика внешних событий.

Формат: ***int E270GetGMSC_ECNT(PE270)***
Назначение: Считать текущее значение счетчика внешних событий и внутреннего таймера модуля. При успешном завершении данной функции обновляются поля gmsc и ecnt
Передаваемые параметры: • указатель на структуру конфигурации модуля.

6.3.5.5. Функции для работы с EEPROM.

Формат: ***int E270WriteEnableEEPROM(PE270)***
Назначение: Разрешить запись в EEPROM модуля/мезонина. При успешном завершении данной функции будет разрешена запись в пользовательскую область EEPROM. Примечание: разрешать запись в EEPROM необходимо непосредственно перед каждым вызовом функции E270WriteEEPROM, иначе запись будет игнорироваться.
Передаваемые параметры: • указатель на структуру конфигурации модуля.

Формат: ***int E270WriteEEPROM(PE270, BYTE)***
Назначение: Произвести записать в EEPROM модуля/мезонина. При успешном завершении данной функции в пользовательскую область EEPROM модуля/мезонина будут записаны данные содержащиеся в соответствующем поле конфигурационной структуры. Примечание: непосредственно перед записью в EEPROM необходимо вызывать функцию E270WriteEnableEEPROM, иначе запись будет игнорироваться.
Передаваемые параметры: • указатель на структуру конфигурации модуля. • номер модуля/мезонина (0-модуль, 1..8 мезонин)

Формат: ***int E270ReadEEPROM(PE270, BYTE)***
Назначение: Произвести чтение EEPROM модуля/мезонина. При успешном завершении данной функции в соответствующем поле конфигурационной структуры будут содержаться данные из EEPROM модуля/мезонина.
Передаваемые параметры: • указатель на структуру конфигурации модуля. • номер модуля/мезонина (0-модуль, 1..8 мезонин)

Формат:	***int E270CRCCheck(PE270, BYTE)***
Назначение: Произвести проверку достоверности данных EEPROM модуля/мезонина находящихся в соответствующем поле конфигурационной структуры. Данная функция проверяет достоверность данных по контрольной сумме. При правильной контрольной сумме возвращаемое значение равно нулю, в противном случае для мезонинов возвращается 1, а для модуля возвращаемое значение интерпретируется так: бит 0 – ошибка кс системной области; бит 1 – ошибка кс области начальной конфигурации модуля; бит 2 – ошибка кс пользовательской области.	
Передаваемые параметры: • указатель на структуру конфигурации модуля. • номер модуля/мезонина (0-модуль, 1..8 мезонин)	

Формат:	***int E270CRCUpdate(PE270, BYTE)***
Назначение: Пересчет контрольной суммы данных EEPROM модуля/мезонина находящихся в соответствующем поле конфигурационной структуры.	
Передаваемые параметры: • указатель на структуру конфигурации модуля. • номер модуля/мезонина (0-модуль, 1..8 мезонин)	

6.3.5.6. Функции вспомогательного характера.

Формат:	***double E270GetValue(PE270, PDATA, BYTE, BYTE)***
Назначение: Получить значение выбранного канала АЦП. Функция возвращает значение выбранного аналогового канала в отсчетах АЦП или в физических единицах измерения соответствующих указанному каналу. (см. также описание IDString).	
Передаваемые параметры: • указатель на структуру конфигурации модуля. • указатель на структуру содержащую данные АЦП • номер канала АЦП (0..15 – 14-ти разрядное АЦП, 16..23 – 8-ми разрядное АЦП) • формат результата (0 – код АЦП, 1 – физические величины)	

Формат:	***const char* E270GetSerialNumber(PE270, BYTE)***
Назначение: Получить серийный номер модуля/мезонина в виде текстовой строки.	
Передаваемые параметры: • указатель на структуру конфигурации модуля. • номер модуля/мезонина (0-модуль, 1..8 мезонин)	

Формат:	***const char* E270GetAPIMessage()***
Назначение: Получить текстовое сообщение о результате выполнения последней вызванной библиотечной функции.	
Передаваемые параметры: нет	

6.4. Демонстрационные и примерные программы.

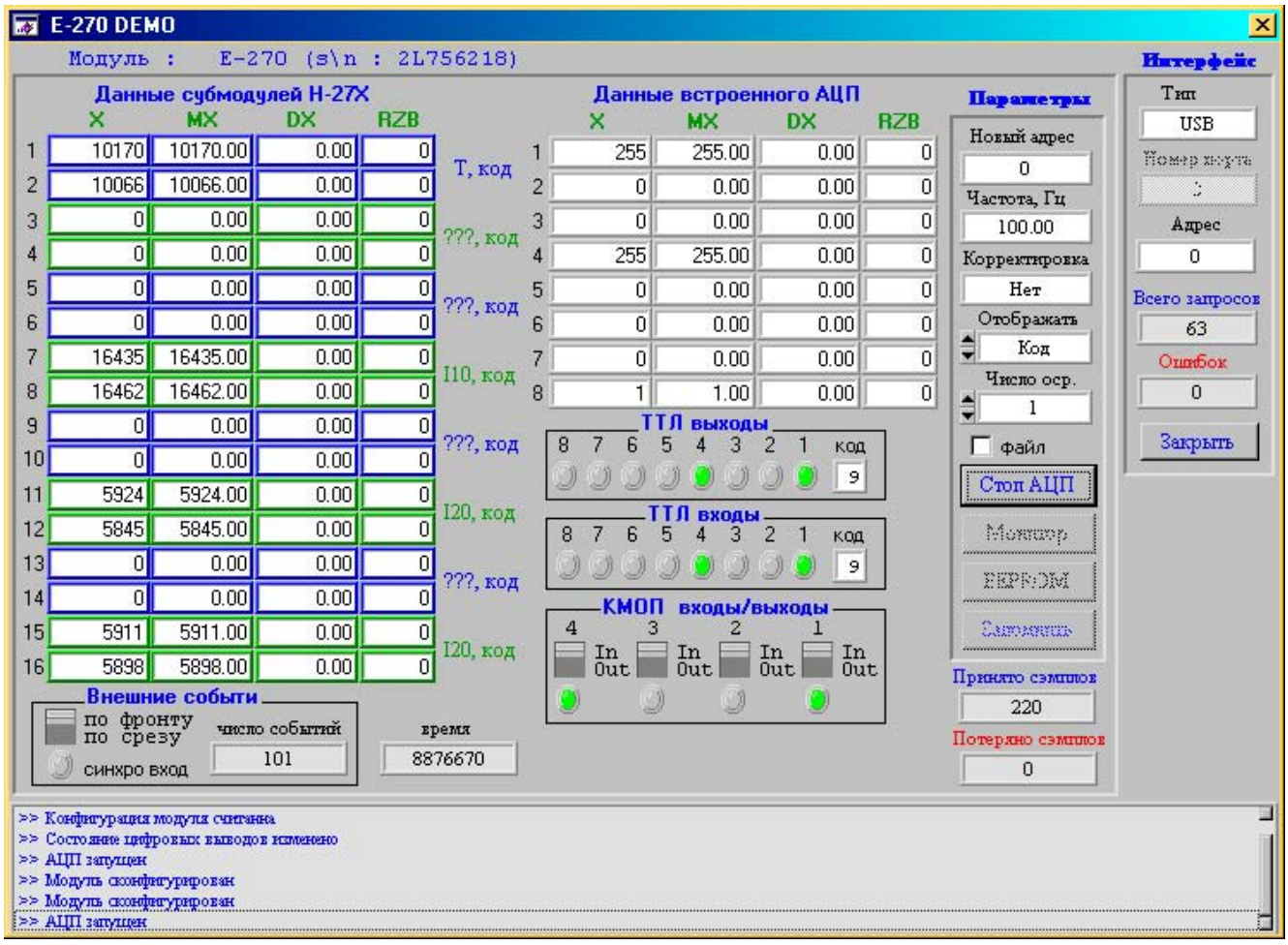
6.4.6. Примерные программы.

В каталогах \E270\EXAMPL.BC, \E270\EXAMPLE.VC и \E270\EXAMPLE.D содержатся примерные программы, написанные под следующие среды разработки: Borland Builder, MS Visual C++ и Borland Delphi соответственно. Все примерные программы идентичны по своим функциональным возможностям и дают простой пример использования библиотечных функций.

6.4.7. Демонстрационные программы.

В каталоге \E270\DEMO находится демонстрационная программа E270DEMO. Эта программа позволяет проверить работоспособность модуля и дает исчерпывающий пример работы с модулем, используя библиотечных функций.

6.4.7.1. Основная панель демонстрационной программы.

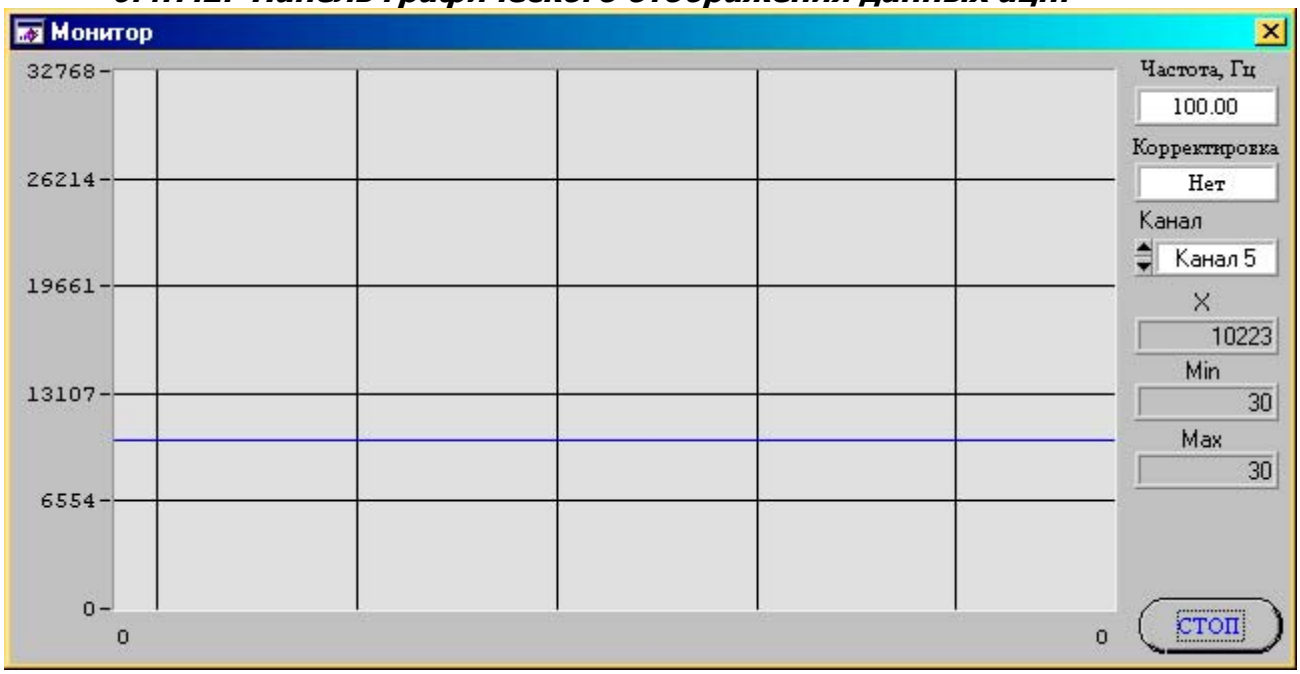


Основная панель демонстрационной программы позволяет отображать и выбирать: тип и параметры интерфейса, конфигурацию модуля, состояние линий цифрового вода/вывода, результаты аналого-цифрового преобразования.

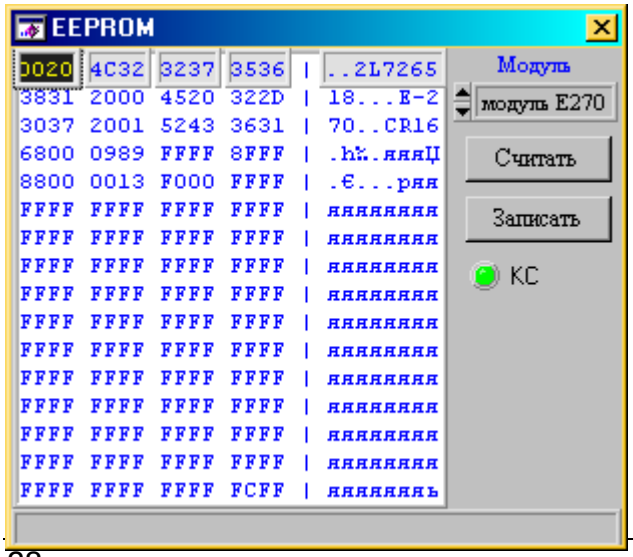
- ***Набор элементов «интерфейс»*** (справа) позволяет задать тип интерфейса, его параметры (номер COM-порта, адрес модуля) и изменить его состояние (открыт/закрыт). Кроме того, здесь же отображается статистика о переданных и обработанных запросах.
- ***Набор элементов «параметры»*** (в центре) позволяет выполнить следующие действия:
  - задать новый адрес модуля
  - задать новую частоту дискретизации
  - включить использование корректирующих коэффициентов внутри модуля
  - изменить формат отображения данных ацп
  - выбрать число осреднений при отображении данных ацп
  - включить режим сохранения данных ацп в текстовый файл
  - начать/остановить процесс отображения данных ацп
  - вызвать панель графического отображения данных ацп
  - вызвать панель отображения/редактирования содержимого EEPROM модуля/мезонина
  - записать текущую конфигурацию модуля в EEPROM
- ***Набор элементов «ТТЛ входы/выходы»*** позволяет задать состояние цифровых ТТЛ выходов модуля и отобразить состояние цифровых ТТЛ входов
- ***Набор элементов «КМОП входы/выходы»*** позволяет задать конфигурацию и состояние дополнительных цифровых КМОП входов/выходов.

- ***Набор элементов «Внешние события»*** позволяет задать параметры счетчика событий: чувствительность счетчика по переднему или заднему фронту импульса, использование внешнего события для синхронизации сбора аналоговых данных. А также отображают текущее значение счетчика событий.
- ***Элемент «Время»*** отображает в миллисекундах текущее значение внутреннего таймера модуля.
- ***Набор элементов, отображающий результаты ацп*** содержит:
  - текущее значение данных ацп
  - усредненное значение данных ацп
  - среднеквадратичное отклонение на интервале усреднения
  - разброс значений на интервале усреднения (на интервале усреднения подсчитывается максимум, минимум и отображается разность)
  - тип мезонина
  - число принятых и потерянных «блоков данных ацп/цвв»
- ***Кнопка «ПАСПОРТ»*** создает файлы необходимые для формирования паспорта модуля(см. папку DEMO\PASPORTA\<серийный номер модуля>)

6.4.7.2. Панель графического отображения данных ацп.



Данная панель позволяет представить данные ацп выбранного канала в графическом виде.



6.4.7.3. Панель отображения/редактирования содержимого EEPROM модуля/мезонина.

Данная панель позволяет отображать, редактировать, считывать и записывать содержимое EEPROM выбранного модуля/мезонина.