Описание библиотеки OWEN_IO

Оглавление

1 Назначение

Библиотека предназначена для связи с приборами ОВЕН, поддерживающими сетевой интерфейс RS-485 и "токовая петля".

По интерфейсу RS-485 предусматривается работа с:

- автоматическими преобразователями RS-232/RS-485, например, АС3-М, ND-6530;
- автоматическими преобразователями USB/RS-485, например, АС4, I-7561;
- полуавтоматическими преобразователями RS-232/RS-485 например, АС3.

По интерфейсу "токовая петля" предусматривается работа с адаптером АС-2.

Список соответствий параметров приборов функциям библиотеки находится в файле tags.txt.

2 Порядок индексации

В функциях, имеющих в качестве входного значения index, передаются следующие значения:

– для оперативных параметров передается значение "-1" – то есть без индекса, индексация каналов осуществляется увеличением базового адреса прибора;

Исключение составляют параметры Pv, LuPv и rEAd в ТРМ2xx v.01.0013 – индексация идет по адресам, но дополнительно передается индекс со значением "0".

– для конфигурационных параметров индексация идет через значение index;

– для параметров, не требующих индексации, значение index = -1.

Конфигурационные параметры могут иметь индекс или не иметь такового. Индекс не требуется, если конфигурационный параметр представлен в приборе в единственном числе.

Индекс оперативного параметра передается в адресе прибора, в поле данных индекс не ставится.

Пример

Для чтения значения параметра rEAd канала N (1…8) ТРМ138 следует добавлять к базовому адресу устройства число N-1. Так, для чтения значения на третьем канале с базовым адресом 24 следует вызвать ReadIEEE32(24+(3-1),0,"rEAd",val,tm,-1). То есть восьмиканальный прибор представляется как 8 независимых одноканальных приборов, каждый со своим базовым адресом.

3 Использование в VB.NET

Типы параметров в описаниях функций даны для Visual Basic 6. При использованиии в VB.NET необходимо учитывать соответствия, приведенные в таблице.

VB	VB.NET	Примечания
Integer	Short	16 бит
Long	Integer	32 бита

4 Описание функций для работы в сети RS-485

4.1 Вспомогательные функции

4.1.1 SetApiMode

Выбрать модель библиотеки.

C++	int SetApiMode(int mode);
Delphi	function SetApiMode(Mode: Integer): Integer;
VB	Function SetApiMode () As Long

Параметры

n

Модель библиотеки.

OWENIO_API_OLD (0)	старая модель
OWENIO_API_NEW (1)	новая модель

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

Примечания

При смене модели библиотеки все открытые порты будут принудительно закрыты. Любые обращения к функциям чтения/записи данных без повторного открытия портов после смены модели недопустимы.

4.1.2 OpenPort

Открыть порт для интерфейса RS-485.

C++	int OpenPort(DWORD n,DWORD speed,DWORD parity,DWORD bits,DWORD stop,DWORD converter);
Delphi	function OpenPort(n,speed, parity,bits,stop,converter:DWORD): Integer;
VB	Function OpenPort(ByVal n As Long, ByVal speed As Long, ByVal parity As Long, ByVal bits As Long, ByVal bitstop As Long, ByVal converter As Long) As Long

Параметры

n

Номер последовательного порта, для COMnn n=nn-1.

speed

Скорость порта.

spd_300 (-3)	300 кбит/с
spd_600 (-2)	600 кбит/с
spd_1200 (-1)	1200 кбит/с
spd_2400 (0)	2400 кбит/с
spd_4800 (1)	4800 кбит/с
spd_9600 (2)	9600 кбит/с
spd_14400 (3)	14400 кбит/с
spd_19200 (4)	19200 кбит/с
spd_28800 (5)	28800 кбит/с
spd_38800 (6)	38800 кбит/с
spd_57600 (7)	57600 кбит/с
spd_115200 (8)	115200 кбит/с

parity

Бит четности.

prty_NONE (0)	Без бита четности
prty_EVEN (1)	Чет
prty_ODD (2)	Нечет

bits

Бит данных.

7 бит	databits_7
8 бит	databits_8

stop

Стоповые биты.

1 стоп-бит	stopbit_1
1,5 стоп-бит	stopbit_1_5
2 стоп-бит	stopbit_2

converter

Управление передатчиком RS-485 от компьютера.

RS485CONV_MANUAL (0)	Сигналом RTS (АС-3 или другой полуавтоматический преобразователь)
RS485CONV_AUTO (1)	Автоматический преобразователь
RS485CONV_MANUAL_DTR (2)	Сигналом DTR (полуавтоматический преобразователь)

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

Примечания

Для приборов ТРМ-151, ТРМ-138, ТРМ-148, ТРМ-133, МВА-8, МВУ-8 являются недопустимыми следующие комбинации числа бит данных, стоповых бит и битов четности:

– 7 бит данных, 1 стоповый бит, отсутствие бита четности;

– 8 бит данных, 2 стоповых бита, бит четности – проверка на четность;

– 8 бит данных, 2 стоповых бита, бит четности – проверка на нечетность.

4.1.3 ClosePort

Освободить порт для интерфейса RS-485. В новой модели – освободить активный порт, занятый интерфейсом RS-485.

C++	int ClosePort();
Delphi	function ClosePort: Integer;
VB	Function ClosePort () As Long

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.1.4 SelectPort

Выбрать активный порт, то есть порт, с которым будут производиться все действия. Только для новой модели.

C++	int SelectPort(DWORD n);
Delphi	function SelectPort(n: DWORD): Integer;
VB	Function SelectPort (ByVal n As Long) As Long

Параметры

n

Номер ранее открытого порта.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.1.5 SetupPort

Настроить порт. Только для новой модели.

C++	int SetupPort(DWORD n,DWORD speed,DWORD

	parity,DWORD bits,DWORD stop,DWORD converter);
Delphi	function SetupPort(n,speed,parity,bits,stop,converter:DWORD): Integer;
VB	Function SetupPort(ByVal n As Long, ByVal speed As Long, ByVal parity As Long, ByVal bits As Long, ByVal bitstop As Long, ByVal converter As Long) As Long

Параметры

n

Номер последовательного порта, для COMnn n=nn-1.

speed

Скорость порта.

spd_300 (-3)	300 кбит/с
spd_600 (-2)	600 кбит/с
spd_1200 (-1)	1200 кбит/с
spd_2400 (0)	2400 кбит/с
spd_4800 (1)	4800 кбит/с
spd_9600 (2)	9600 кбит/с
spd_14400 (3)	14400 кбит/с
spd_19200 (4)	19200 кбит/с
spd_28800 (5)	28800 кбит/с
spd_38800 (6)	38800 кбит/с
spd_57600 (7)	57600 кбит/с
spd_115200 (8)	115200 кбит/с

parity

Бит четности.

prty_NONE (0)	отсутствует
prty_EVEN (1)	проверка на четность
prty_ODD (2)	проверка на нечетность

bits

Бит данных.

7 бит	databits_7
8 бит	databits_8

stop

Стоповые биты.

1 стоп-бит	stopbit_1
1,5 стоп-бит	stopbit_1_5
2 стоп-бит	stopbit_2

converter

Управление передатчиком RS-485 от компьютера.

RS485CONV_MANUAL (0)	Сигналом RTS (АС-3 или другой полуавтоматический преобразователь)
RS485CONV_AUTO (1)	Автоматический преобразователь
RS485CONV_MANUAL_DTR (2)	Сигналом DTR (полуавтоматический преобразователь)

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

Примечания

Для приборов ТРМ-151, ТРМ-138, ТРМ-148, ТРМ-133, МВА-8, МВУ-8 являются недопустимыми следующие комбинации числа бит данных, стоповых бит и битов четности:

– 7 бит данных, 1 стоповый бит, отсутствие бита четности;

– 8 бит данных, 2 стоповых бита, бит четности – проверка на четность;

– 8 бит данных, 2 стоповых бита, бит четности – проверка на нечетность.

4.1.6 LastErrToStr

Помещает описание последней ошибки в буфер.

C++	int LastErrToStr(char *res);
Delphi	procedure LastErrToStr(res:PChar);
VB	Function LastErrToStr(ByVal res As String) As Long

Параметры

res

Указатель на буфер размером не менее 300 символов (выделяется пользовательской программой).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

Примечания

Существует три вида ошибок: ERR_NERR, ERR_DEVERR и внутренние ошибки библиотеки.

Если метод чтения/записи параметра возвратил ошибку ERR_NERR, в буфер помещается описание сетевой ошибки nErr, которую вернул прибор. В библиотеке используются следующие коды и описания ошибок:

Код	Описание

0	ошибки обмена отсутствуют
1	устройство не отвечает
2	задано положение десятичной точки, превышающее 3
3	попытка модификации ROM-параметра
4	не целое число при записи индекса строки или времени
5	неверно задано положение точки (при фиксированной точке)
6	значение мантиссы превышает ограничения дескриптора
7	попытка редактирования атрибута пользователем, не являющимся владельцем параметра
8	у запрошенного параметра отсутствуют признаки
33	аппаратная ошибка кадрирования
34	ошибка в 8-ом бите посылки
35	ошибка в 9-ом бите посылки
36	ошибка приема стоп-байта (стоп пришел не вовремя)
37	переполнение буфера
38	принят недопустимый символ
39	неверная контрольная сумма
40	не найден дескриптор
41	не найдена сетевая функция, хотя дескриптор найден
48	мантисса двоично-десятичного параметра содержит ошибку
49	размер поля данных не соответствует ожидаемому
50	значение бита запроса не соответствует ожидаемому
51	редактирование параметра запрещено индивидуальным настройками
52	недопустимо большой линейный индекс
53	индекс параметра превышает ограничитель индекса
54	индекс параметра превышает ограничитель индекса
55	данный код не используется
56	запрещающий групповой атрибут находится на уровне 0 (в корне)
57	запрещающий групповой атрибут находится на уровне 1
58	запрещающий групповой атрибут находится на уровне 2
59	запрещающий групповой атрибут находится на уровне 3
60	запрещающий групповой атрибут находится на уровне 4
61	запрещающий групповой атрибут находится на уровне 5
62	запрещающий групповой атрибут находится на уровне 6
63	запрещающий групповой атрибут находится на уровне 7
65	выполняется другая задача (Сегмент COMMON занят)
66	задача еще не запущена(Сегмент COMMON свободен)
67	запрошенная задача уже выполняется
68	программе неизвестна запрошенная функция
69	в программе стоит заглушка функции WhatCOMState()
70	в программе стоит заглушка функции RunCOMTask()

Если метод чтения/записи параметра возвратил ошибку ERR_DEVERR, прибор вернул признак исключительной ситуации. В буфер помещается описание этой исключительной ситуации. Ее код можно получить методом **GetExtendedLastErr()**.

Библиотека распознает исключительные ситуации, перечисленные в п. 5.3.

Если произошла внутренняя ошибка библиотеки, возвращается ее описание, см. п. 5.1, 5.2.

4.1.7 GetExtendedLastErr

Возвращает код исключительной ситуации.

C++	int GetExtendedLastErr();
Delphi	function GetExtendedLastErr(): Integer;
VB	Function GetExtendedLastErr() As Long

Возвращаемое значение
Код исключительной ситуации.

4.1.8 SetDbgIndication

Настроить вывод в консоль отладочной информации об обмене в сети RS-485.

C++	int SetDbgIndication(int);
Delphi	function SetDbgIndicator(ind: Integer): Integer;
VB	Function SetDbgIndicator(ind As integer) As Long

Настройка по умолчанию – не выводить отладочную информацию.

Предопределенные константы маски вывода в owen_io.h

SHOW_FRAMES (1)	Выводить отправленные и полученные ASCII-кадры
SHOW_SEND_DATA (2)	Выводить отправленные двоичные кадры
SHOW_RECV_DATA (4)	Выводить принятые двоичные кадры
SHOW_RECV_ERRORS (16)	Выводить информацию об ошибках приема
SHOW_RECV_ALL (32)	Выводить все принятые из COM-порта данные

Внимание! Для просмотра индикации в оконных приложениях следует сначала вызвать функцию **AllocConsole()**.

Возвращаемое значение
При успехе ERR_OK, при неудаче <0.

4.1.9 GetMaxRetriesGlobal

Получить число повторных попыток связи при ошибке обмена с прибором.

C++	int GetMaxRetriesGlobal(void);
Delphi	function GetMaxRetriesGlobal(): Integer;
VB	Function GetMaxRetriesGlobal() As Long

Число перезапросов по умолчанию – 1.

Возвращаемое значение
Число перезапросов.

4.1.10 SetMaxRetriesGlobal

Установить число повторных попыток связи при ошибке обмена с прибором.

C++	int SetMaxRetriesGlobal(int NumberOfRetries);
Delphi	function SetMaxRetriesGlobal(NumberOfRetries: Integer): integer;
VB	Function SetMaxRetriesGlobal(NumberOfRetries As Integer) As Long

Параметры

NumberOfRetries
Число перезапросов при ошибке обмена с прибором.

Число перезапросов по умолчанию – 1.

Возвращаемое значение
При успехе ERR_OK (предыдущее значение), при неудаче <0.

4.2 Чтение целочисленных параметров (см. п. 5.1.4 "Описание протокола обмена между ПЭВМ и приборами ОВЕН")

4.2.1 ReadSInt

Чтение целочисленного значения со знаком.

C++	int ReadSInt(DWORD adr,DWORD adr_type,char *command,int &value, int index);

Delphi	function ReadSInt(adr,adr_type:DWORD;command:PChar;var value:Integer;index:Integer): Integer;
VB	Function ReadSInt(ByVal adr As Long, ByVal adr_type As Long, ByVal cmd As String, res As Long, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.2.2 ReadUInt

Чтение беззнакового целочисленного значения.

C++	int ReadUInt(DWORD adr,DWORD adr_type,char *command,unsigned int &value, int index);
Delphi	function ReadUInt(adr_type,adr:DWORD;command:PChar;var value:DWORD;index:Integer): Integer;
VB	Function ReadUInt(ByVal adr As Long, ByVal adr_type As Long, ByVal cmd As String, res As Long, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command
Команда.

value
Значение считанного параметра.

index
Индекс (см. п. 2).

Возвращаемое значение
При успехе ERR_OK, при неудаче <0.

4.3 Запись целочисленных параметров (см. п. 5.1.4 "Описание протокола обмена между ПЭВМ и приборами ОВЕН")

4.3.1 WriteByte

Запись целочисленного значения размером один байт.

C++	int WriteByte(DWORD adr,DWORD adr_type,char *command,int value,int index);
Delphi	function WriteByte(adr,adr_type:DWORD;command:PChar;value:Integer;index:Integer): Integer;
VB	Function WriteByte(ByVal adr As Long, ByVal adr_type As Long, ByVal cmd As String, ByVal value As Long, ByVal Index As Integer) As Long

Параметры

adr
Адрес устройства.

adr_type
Длина адреса.

0	8 бит
1	11 бит

command
Команда.

value
Значение считанного параметра.

index
Индекс (см. п. 2).

Возвращаемое значение
При успехе ERR_OK, при неудаче <0.

4.3.2 WriteWord

Запись целочисленного значения размером два байта.

C++	int WriteWord(DWORD adr,DWORD adr_type,char *command,int value,int index);
Delphi	function WriteWord(adr,adr_type:DWORD;command:PChar;value :Integer;index:Integer): Integer;
VB	Function WriteWord(ByVal adr As Long, ByVal adr_type As Long, ByVal cmd As String, ByVal value As Long, ByVal Index As Integer) As Long

Параметры
adr
Адрес устройства.

adr_type
Длина адреса.

0	8 бит
1	11 бит

command
Команда.

value
Значение считанного параметра.

index
Индекс (см. п. 2).

Возвращаемое значение
При успехе ERR_OK, при неудаче <0.

4.4 Чтение значений с плавающей точкой

4.4.1 ReadFloat24

Чтение значения с плавающей точкой в формате PIC (см.п. 5.1.1 "Описание протокола обмена между ПЭВМ и приборами ОВЕН").

C++	int ReadFloat24(DWORD adr,DWORD adr_type,char *command,float &value, int index);
Delphi	function ReadFloat24(adr,adr_type:DWORD;command:PChar;var value:Single;index:Integer):Integer;
VB	Function ReadFloat24(ByVal adr As Long, ByVal adr_type As Long, ByVal cmd As String, res As Single, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.4.2 ReadStoredDotS

Чтение значения с десятичной точкой в формате ОВЕН (см. п. 5.1.2 "Описание протокола обмена между ПЭВМ и приборами ОВЕН").

C++	int ReadStoredDotS(DWORD adr,DWORD adr_type,char *command,float &value, int index);
Delphi	function ReadStoredDotS (adr,adr_type:DWORD;command:PChar; var value:Single;index:Integer):Integer;
VB	Function ReadStoredDotSng(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, res As Single, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.4.3 ReadStoredDotEx

Чтение значения с десятичной точкой в формате ОВЕН (см. п. 5.1.2 "Описание протокола обмена между ПЭВМ и приборами ОВЕН").

C++	int ReadStoredDotEx(DWORD adr,DWORD adr_type,char *command,float &value, int& point, int index);
Delphi	function ReadStoredDotEx (adr,adr_type:DWORD;command:PChar; var value:Single;var point:Integer;index:Integer):Integer;
VB	Function ReadStoredDotEx(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, res As Single, Point As Integer, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

point

Положение десятичной точки, считанное из прибора.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.4.4 ReadIEEE32

Чтение значения с плавающей точкой в формате IEEE32 (см. п. 5.1.6 "Описание протокола обмена между ПЭВМ и приборами ОВЕН").

C++	int ReadIEEE32(DWORD adr,DWORD adr_type,char *command,float &value,int &time,int index);
Delphi	function ReadIEEE32(adr,adr_type:DWORD;command:PChar;var value:Single;var time:Integer;index:Integer):Integer;
VB	Function ReadIEEE32(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, res As Single, time As Long, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

time

Время измерения (см. п. 5.1.6 "Описание протокола обмена между ПЭВМ и приборами ОВЕН").

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.5 Запись значений с плавающей точкой

4.5.1 WriteFloat24

Запись значения с плавающей точкой в формате PIC (см.п. 5.1.1 "Описание протокола обмена между ПЭВМ и приборами ОВЕН").

C++	int WriteFloat24(DWORD adr,DWORD adr_type,char *command,float value, int index);
Delphi	function WriteFloat24(adr,adr_type:DWORD;command:PChar;value:Single;index:Integer): Integer;
VB	Function WriteFloat24(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, ByVal value As Single, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.5.2 WriteStoredDotS

Запись значения с десятичной точкой в формате ОВЕН (см. п. 5.1.2 "Описание протокола обмена между ПЭВМ и приборами ОВЕН"). Положение десятичной точки вычисляется автоматически.

Некоторые приборы требуют явного задания положения десятичной точки. В этом случае используйте функцию *WriteStoredDotEx.*

C++	int WriteStoredDotS(DWORD adr,DWORD adr_type,char *command,float value, int index);
Delphi	function WriteStoredDotS(adr,adr_type:DWORD;command:PChar; value:Single;index:Integer): Integer;
VB	Function WriteStoredDotSng(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, ByVal value As Single, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.5.3 WriteStoredDotEx

Запись значения с десятичной точкой в формате ОВЕН (см. п. 5.1.2 "Описание протокола обмена между ПЭВМ и приборами ОВЕН") с указанием положения десятичной точки.

C++	int WriteStoredDotEx(DWORD adr,DWORD adr_type,char *command,float value, int point, int index);
Delphi	function WriteStoredDotEx(adr,adr_type:DWORD;command:PChar; value:Single;point:Integer;index:Integer): Integer;
VB	Function WriteStoredDotEx(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, ByVal value As Single, ByVal point As Integer, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

point

Положение десятичной точки.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.5.4 WriteIEEE32

Запись значения с плавающей точкой в формате IEEE32 (см.п. 5.1.1 "Описание протокола обмена между ПЭВМ и приборами ОВЕН").

C++	int WriteIEEE32(DWORD adr,DWORD adr_type,char *command,float value,int index);
Delphi	function WriteIEEE32(adr,adr_type:DWORD;command:PChar;var value:Single;index:Integer):Integer;
VB	Function WriteIEEE32(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, ByVal res As Single, ByVal Index As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

index

Индекс (см. п. 2).

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.6 Универсальная функция для обмена с устройствами, поддерживающими протокол ОВЕН

4.6.1 OwenIO

C++	int OwenIO(DWORD adr,DWORD adr_type,DWORD is_read,char *command,char *params,DWORD *param_sz);
Delphi	function OwenIO(adr,adr_type,is_read:DWORD;command,params:PChar; var param_sz:Integer): Integer;
VB	Function OwenIO(ByVal adr As Long, ByVal adr_type As Long, ByVal is_read As Long, ByVal command As String, ByVal params As String, param_sz As Long) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

is_read

Чтение/запись данных.

0	запись данных в устройство
1	чтение данных из устройства

command

Команда.

params

Параметры, передаваемые/получаемые в/из устройства.

param_sz

Размер параметра, передается и возвращается.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.7 Специфические функции для различных устройств ОВЕН

4.7.1 ReadDTMR

Чтение значения параметра DTMR устройства СИ8

C++	int ReadDTMR(DWORD adr,DWORD adr_type,int &hrs,int &mins,int &sec,int &msec);
Delphi	function ReadDTMR(adr,adr_type:DWORD;var hrs:Integer;var mins:Integer;var sec:Integer;var msec:Integer):Integer;
VB	Function ReadDTMR(ByVal adr As Long, ByVal adr_type As Long, a_hrs As Long, a_mins As Long, a_sec As Long, a_msec As Long) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

hrs

Считанное значение счетчика часов.

mins

Считанное значение счетчика минут.

sec

Считанное значение счетчика секунд.

msec

Считанное значение счетчика миллисекунд.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.7.2 ReadSI8BCD

Чтение значений параметров DCNT и DSPD устройства СИ8

C++	int ReadSI8BCD(DWORD adr,DWORD adr_type,char *command,int &value);
Delphi	function ReadSI8BCD(adr,adr_type:DWORD;command:PChar; var value:Integer):Integer;
VB	Function ReadSI8BCD(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, value As Integer) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.7.3 WriteCSET

Запись команды CSET в устройство ПКП1.

C++	int WriteCSET(DWORD adr,DWORD adr_type,int prc);
Delphi	function WriteCSET(adr,adr_type:DWORD;prc:Integer):Integer;
VB	Function WriteCSET(ByVal adr As Long, ByVal adr_type As Long, ByVal prc As Long) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

prc

Записываемое значение в десятых долях процента. Например, для 12,3% prc = 123.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.7.4 ReadPKPBCD

Чтение значений параметров DPRC, DCUR, DTME устройства ПКП1.

C++	int ReadPKPBCD(DWORD adr,DWORD adr_type,char *command,float &value);
Delphi	function ReadPkpBCD(adr,adr_type:DWORD;command:PChar; var value:Single):Integer;
VB	Function ReadPkpBCD(ByVal adr As Long, ByVal adr_type As Long, ByVal command As String, value As Single) As Long

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

command

Команда.

value

Значение считанного параметра.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.7.5 ReadRTC

Чтение значения параметра RTC прибора ТРМ133.

C++	int ReadRTC(DWORD adr,DWORD adr_type,char *result);

Delphi	
VB	

Параметры

adr

Адрес устройства.

adr_type

Длина адреса.

0	8 бит
1	11 бит

result

Показания часов реального времени (время и дата). Буфер размером не менее 30 символов выделяется пользовательской программой.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.8 Прослушивание линии (пассивный режим)

4.8.1 Listen

Прослушивание линии.

C++	int Listen(int timeout, unsigned short *rcv_hash, int *rcv_flag, int *rcv_adr, char *param_data, int *param_sz);
Delphi	function Listen(timeout: Integer; var rcv_hash: WORD; var rcv_flag, rcv_adr: Integer; param_data: PChar; var param_sz: Integer):Integer;
VB	Function Listen(ByVal timeout As Long, rcv_hash As Integer, rcv_flag As Long, rcv_adr As Long, param_data As String, param_sz As Long) As Long

Параметры

timeout

Время в миллисекундах, по истечении которого завершится работа функции в случае, если кадр не принят.

rcv_hash

Указатель на переменную, в которую будет записан хеш-код команды принятого кадра.

rcv_flag

Указатель на переменную, в которую будет записан признак удаленного запроса кадра (см п.3.3.2 "Описание протокола обмена между ПЭВМ и приборами ОВЕН").

rcv_adr

Указатель на переменную, в которую будет записан полный адрес кадра (8-ми битная старшая часть адреса и 3-х битовое расширение адреса узла сети).

param_data

Буфер, в который будет записано содержимое поля данных кадра. Буфер размером не менее 15 байт выделяется пользовательской программой.

param_sz

Указатель на переменную, в которую будет записан размер поля данных кадра.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

4.9 Получение данных из принятого кадра

4.9.1 GetBufferSInt

Получение знакового целочисленного значения.

C++	int GetBufferSInt(unsigned char *buffer, int buffer_size, int *result);
Delphi	function GetBufferSInt(buffer: Pchar; buffer_size: Integer; var result: Integer): Integer;
VB	Function GetBufferSInt(buffer As String, ByVal buffer_size As Integer, result As Integer) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.2 GetBufferUInt

Получение беззнакового целочисленного значения.

C++	int GetBufferUInt(unsigned char *buffer, int buffer_size, unsigned *result);
Delphi	function GetBufferUInt(buffer: Pchar; buffer_size: Integer; var result: DWORD): Integer;
VB	Function GetBufferUInt(buffer As String, ByVal buffer_size As Integer, result As Long) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.3 GetBufferByte

Получение целочисленного значения размером байт.

C++	int GetBufferByte(unsigned char *buffer, int buffer_size, byte *result);
Delphi	function GetBufferByte(buffer: Pchar; buffer_size: Integer; var result: BYTE): Integer;
VB	Function GetBufferByte(buffer As String, ByVal buffer_size As Integer, result As Long) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.4 GetBufferWord

Получение целочисленного значения размером два байта.

C++	int GetBufferWord(unsigned char *buffer, int buffer_size, word *result);
Delphi	function GetBufferWord(buffer: Pchar; buffer_size: Integer; var result: WORD): Integer;
VB	Function GetBufferWord(buffer As String, ByVal buffer_size As Integer, result As Long) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.5 GetBufferFloat24

Получение значения с плавающей точкой в формате PIC.

C++	int GetBufferFloat24(unsigned char *buffer, int buffer_size, float *result);
Delphi	function GetBufferFloat24(buffer: Pchar; buffer_size: Integer; var result: Single): Integer;
VB	Function GetBufferFloat24(buffer As String, ByVal buffer_size As Integer, result As Single) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.6 GetBufferStoredDotS

4.9.7 GetBufferStoredDotU

4.9.8 GetBufferStoredIEEE32

Получение значения с плавающей точкой

C++	int GetBufferStoredDotS(unsigned char *buffer, int buffer_size, float *result);
	int GetBufferStoredDotU(unsigned char *buffer, int buffer_size, float *result);
	int GetBufferStoredIEEE32(unsigned char *buffer, int buffer_size, float *result);

Delphi	function GetBufferStoredDotS(buffer: Pchar; buffer_size: Integer; var result: Single): Integer;
	function GetBufferStoredDotU(buffer: Pchar; buffer_size: Integer; var result: Single): Integer;
	function GetBufferStoredIEEE32(buffer: Pchar; buffer_size: Integer; var result: Single): Integer;

VB	Function GetBufferStoredDotS(buffer As String, ByVal buffer_size As Integer, result As Single) As Long Function GetBufferStoredDotU(buffer As String, ByVal buffer_size As Integer, result As Single) As Long Function GetBufferIEEE32(buffer As String, ByVal buffer_size As Integer, result As Single) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.9 GetBufferDTMR

Получение значения параметра DTMR устройства СИ8.

C++	int GetBufferDTMR(unsigned char *buffer, int buffer_size, int *hrs, int *mins, int *sec, int *msec);
Delphi	function GetBufferDTMR(buffer: Pchar; buffer_size: Integer; var hrs, mins, sec, msec: Integer): Integer;
VB	Function GetBufferDTMR(buffer As String, ByVal buffer_size As Integer, hrs As Integer, mins As Integer, secs As Integer, msecs As Integer) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

hrs

Указатель на переменную, куда будет записано значение счетчика часов.

mins

Указатель на переменную, куда будет записано значение счетчика минут.

secs

Указатель на переменную, куда будет записано значение счетчика секунд.

msecs

Указатель на переменную, куда будет записано значение счетчика миллисекунд.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.10 GetBufferSI8BCD

Получение значений параметров DCNT и DSPD устройства СИ8.

C++	int GetBufferSI8BCD(unsigned char *buffer, int buffer_size, int *result);
Delphi	function GetBufferSI8BCD(buffer: Pchar; buffer_size: Integer; var result: Integer): Integer;
VB	Function GetBufferSI8BCD(buffer As String, ByVal buffer_size As Integer, result As Integer) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.11 GetBufferPkpBCD

Получение значений параметров DPRC,DCUR,DTME устройства ПКП1

C++	int GetBufferPkpBCD(unsigned char *buffer, int buffer_size, float *result);
Delphi	function GetBufferPkpBCD(buffer: Pchar; buffer_size: Integer; var result: Single): Integer;
VB	Function GetBufferPkpBCD(buffer As String, ByVal buffer_size As Integer, result As Single) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

4.9.12 GetBufferRTC

Получение значения RTC прибора ТРМ133.

C++	int GetBufferRTC(unsigned char *buffer, int buffer_size, char *result);
Delphi	function GetBufferRTC(buffer: Pchar; buffer_size: Integer; result: PChar): Integer;
VB	Function GetBufferRTC(buffer As String, ByVal buffer_size As Integer, result As String) As Long

Параметры

buffer

Буфер, в котором содержатся данные кадра.

buffer_size

Размер буфера кадра.

result

Указатель на переменную, куда будут записаны декодированные данные.

Возвращаемое значение

При успехе >=0 (значение указывает размер блока, который был занят переменной), при неудаче <0.

5 Функции для преобразователя интерфейса АС2 (сеть "токовая петля")

5.1 Вспомогательные функции

5.1.1 AC2_Open

Открыть порт для преобразователя интерфейса АС2.

C++	int AC2_Open(DWORD n);
Delphi	function AC2_Open(n:DWORD):Integer;
VB	Function AC2_Open (ByVal n As Long) As Long

Параметры

n

Номер последовательного порта, для COMnn n=nn-1.

Возвращаемое значение
При успехе ERR_OK, при неудаче <0.

5.1.2 AC2_Close

Освободить порт для интерфейса преобразователя интерфейса АС2.

C++	int AC2_Close();
Delphi	function AC2_Close:Integer;
VB	Function AC2_Close () As Long

Возвращаемое значение
При успехе ERR_OK, при неудаче <0.

5.2 Функции работы с устройствами, подключаемыми через АС2

5.2.1 AC2_ReadMpr51

Чтение данных из устройства МПР51.

C++	int AC2_ReadMpr51(DWORD ch,DWORD speed,float &t_prod,float &t_suhogo,float &t_vlag,float &otn_vlag);
Delphi	function AC2_ReadMpr51(ch,speed:DWORD;var t_prod:Single;var t_suhogo:Single;var t_vlag:Single;var otn_vlag:Single):Integer;
VB	Function AC2_ReadMpr51 (ByVal ch As Long, ByVal speed As Long, t_prod As Single, t_suhogo As Single, t_vlag As Single, otn_vlag As Single) As Long

Параметры

ch
Номер линии АС2 (от 0 до 7).

speed
Скорость обмена, возможные значения см. в Руководстве по эксплуатации на соответствующее устройство.

t_prod
Возвращенное значение температуры продукта.

t_suhogo
Возвращенное значение температуры сухого термометра.

t_vlag

Возвращенное значение температуры влажного термометра.

otn_vlag

Возвращенное значение относительной влажности.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

5.2.2 AC2_ReadTRM__PiC

Не поддерживается в библиотеке версии 1.2

Чтение данных из устройств ТРМ серии PiC.

C++	int AC2_ReadTRM__PiC(DWORD ch,float &Temperature,int &Rele1,int &Rele2,float &Ust1,float &Ust2,float &Delta1,float &Delta2);
Delphi	function AC2_ReadTRM__PiC(ch:DWORD; var Temperature:Single;var Rele1:Integer;var Rele2:Integer;var Ust1:Single;var Ust2:Single;var Delta1:Single;var Delta2:Single):Integer;
VB	Function AC2_ReadTRM__PiC (ByVal ch As Long, Temperature As Single, Rele1 As Long, Rele2 As Long, Ust1 As Single, Ust2 As Single, Delta1 As Single, Delta2 As Single) As Long

Параметры

ch

Номер линии AC2 (от 0 до 7).

Temperature

Возвращенное значение температуры, либо -300.0 при ошибках измерения.

Rele1

Возвращенное значение реле № 1.

0	разомкнуто
1	замкнуто

Rele2

Возвращенное значение реле № 2.

0	разомкнуто
1	замкнуто

Ust1

Возвращенное значение уставки 1, либо -300.0 при ошибках измерения.

Ust2

Возвращенное значение уставки 2 , либо -300.0 при ошибках измерения.

Delta1

Возвращенное значение дельты 1, либо -300.0 при ошибках измерения.

Delta2

Возвращенное значение дельты 2 , либо -300.0 при ошибках измерения.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

5.2.3 AC2_WriteTRM__PiC

Не поддерживается в библиотеке версии 1.2

Запись данных в устройства ТРМ серии PiC.

C++	int AC2_WriteTRM__PiC(DWORD ch,float &Temperature,int &Rele1,int &Rele2,float &Ust1,float &Ust2,float &Delta1,float &Delta2);
Delphi	AC2_WriteTRM__PiC(ch:DWORD;ctrl_mask:Integer;Rele1:Integer;Rele2:Integer;Ust1:Single;Ust2:Single;Delta1:Single;Delta2:Single):Integer;
VB	Function AC2_WriteTRM__PiC (ByVal ch As Long, ByVal ctrl_mask As Long, ByVal Rele1 As Long, ByVal Rele2 As Long, ByVal Ust1 As Single, ByVal Ust2 As Single, ByVal Delta1 As Single, ByVal Delta2 As Single) As Long

Параметры

ch

Номер линии AC2 (от 0 до 7).

ctrl_mask

Маска, в которой каждый бит определяет необходимость записи соответствующего параметра: бит 0 =Rele1,бит 1 =Rele2 и т.д.; при нулевом значении бита запись не производится.

Rele1

Значение реле № 1.

0	разомкнуто
1	замкнуто

Rele2

Значение реле № 2.

0	разомкнуто
1	замкнуто

Ust1

Значение уставки 1.

Ust2

Значение уставки 2.

Delta1

Значение дельты 1.

Delta2

Значение дельты 2.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

5.2.4 AC2_ReadTRM_UKT_38_T_and_U

Чтение данных из устройств ТРМ38 и УКТ38В.

C++	int AC2_ReadTRM_UKT_38_T_and_U(DWORD ch, float &t1, float &t2, float &t3, float &t4, float &t5, float &t6, float &t7, float &t8, float &u1, float &u2, float &u3, float &u4, float &u5, float &u6, float &u7, float &u8);
Delphi	function AC2_ReadTRM_UKT_38_T_and_U(ch:DWORD;var t1:Single;var t2:Single;var t3:Single;var t4:Single;var t5:Single;var t6:Single;var t7:Single;var t8:Single;var u1:Single;var u2:Single;var u3:Single;var u4:Single;var u5:Single;var u6:Single;var u7:Single;var u8:Single):Integer;

VB	Function AC2_ReadTRM_UKT_38_T_and_U (ByVal ch As Long, t1 As Single, t2 As Single, t3 As Single, t4 As Single, t5 As Single, t6 As Single, t7 As Single, t8 As Single, u1 As Single, u2 As Single, u3 As Single, u4 As Single, u5 As Single, u6 As Single, u7 As Single, u8 As Single) As Long

Параметры

ch

Номер линии AC2 (от 0 до 7).

t(X)

Возвращаемое значение температуры на канале (X), или -300.0 при ошибках измерения.

u(X)

Возвращаемое значение уставки на канале (X), или -300.0 при ошибках измерения.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

5.2.5 AC2_ReadUKT38sh4_IU

Чтение данных из устройства УКТ38 Щ4 – ток и напряжение.

C++	int AC2_ReadUKT38sh4_IU(DWORD ch,DWORD speed, float &t1, float &t2, float &t3, float &t4, float &t5, float &t6, float &t7, float &t8, float &u1, float &u2, float &u3, float &u4, float &u5, float &u6, float &u7, float &u8, float &d1, float &d2, float &d3, float &d4,

	float &d5, float &d6, float &d7, float &d8);
Delphi	function AC2_ReadUKT38sh4_IU(ch:DWORD;speed:DWORD; var t1:Single;var t2:Single;var t3:Single;var t4:Single;var t5:Single;var t6:Single;var t7:Single;var t8:Single;var u1:Single;var u2:Single;var u3:Single;var u4:Single;var u5:Single;var u6:Single;var u7:Single;var u8:Single;var d1:Single;var d2:Single;var d3:Single;var d4:Single;var d5:Single;var d6:Single;var d7:Single;var d8:Single):Integer;
VB	Function AC2_ReadUKT38sh4_IU (ByVal ch As Long, ByVal speed As Long, t1 As Single, t2 As Single, t3 As Single, t4 As Single, t5 As Single, t6 As Single, t7 As Single, t8 As Single, u1 As Single, u2 As Single, u3 As Single, u4 As Single, u5 As Single, u6 As Single, u7 As Single, u8 As Single, d1 As Single, d2 As Single, d3 As Single, d4 As Single, d5 As Single, d6 As Single, d7 As Single, d8 As Single) As Long

Параметры

ch

Номер линии AC2 (от 0 до 7).

speed

Скорость обмена, возможные значения см. в Руководстве по эксплуатации на соответствующее устройство.

t(X)

Возвращаемое значение температуры канала (X), или -300.0 при ошибках измерения.

u(X)

Возвращаемое значение уставки канала (X), или -300.0 при ошибках измерения.

d(X)

Возвращаемое значение дельты канала (X), или -300.0 при ошибках измерения.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

5.2.6 AC2_ReadUKT38sh4_res

Чтение данных из устройства УКТ38 Щ4-ТС.

C++	int AC2_ReadUKT38sh4_res(DWORD ch,DWORD speed,

	float &t1, float &t2, float &t3, float &t4, float &t5, float &t6, float &t7, float &t8, float &u1, float &u2, float &u3, float &u4, float &u5, float &u6, float &u7, float &u8, float &d1, float &d2, float &d3, float &d4, float &d5, float &d6, float &d7, float &d8);
Delphi	function AC2_ReadUKT38sh4_res(ch:DWORD;speed:DWORD; var t1:Single;var t2:Single;var t3:Single;var t4:Single;var t5:Single;var t6:Single;var t7:Single;var t8:Single;var u1:Single;var u2:Single;var u3:Single;var u4:Single;var u5:Single;var u6:Single;var u7:Single;var u8:Single;var d1:Single;var d2:Single;var d3:Single;var d4:Single;var d5:Single;var d6:Single;var d7:Single;var d8:Single):Integer;
VB	Function AC2_ReadUKT38sh4_res (ByVal ch As Long, ByVal speed As Long, t1 As Single, t2 As Single, t3 As Single, t4 As Single, t5 As Single, t6 As Single, t7 As Single, t8 As Single, u1 As Single, u2 As Single, u3 As Single, u4 As Single, u5 As Single, u6 As Single, u7 As Single, u8 As Single, d1 As Single, d2 As Single, d3 As Single, d4 As Single, d5 As Single, d6 As Single, d7 As Single, d8 As Single) As Long

Параметры

ch

Номер линии AC2 (от 0 до 7).

speed

Скорость обмена, возможные значения см. в Руководстве по эксплуатации на соответствующее устройство.

t(X)

Возвращаемое значение температуры канала (X), или -300.0 при ошибках измерения.

u(X)

Возвращаемое значение уставки канала (X), или -300.0 при ошибках измерения.

d(X)

Возвращаемое значение дельты канала (X), или -300.0 при ошибках измерения.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

5.2.7 AC2_ReadUKT38sh4_trp

Чтение данных из устройства УКТ38 Щ4-ТП, УКТ38 Щ4-ТПП.

C++	int AC2_ReadUKT38sh4_trp(DWORD ch,DWORD speed, float &t1, float &t2, float &t3, float &t4, float &t5, float &t6, float &t7, float &t8, float &u1, float &u2, float &u3, float &u4, float &u5, float &u6, float &u7, float &u8, float &d1, float &d2, float &d3, float &d4, float &d5, float &d6, float &d7, float &d8);
Delphi	function AC2_ReadUKT38sh4_trp(ch:DWORD;speed:DWORD;

	var t1:Single;var t2:Single;var t3:Single;var t4:Single;var t5:Single;var t6:Single;var t7:Single;var t8:Single;var u1:Single;var u2:Single;var u3:Single;var u4:Single;var u5:Single;var u6:Single;var u7:Single;var u8:Single;var d1:Single;var d2:Single;var d3:Single;var d4:Single;var d5:Single;var d6:Single;var d7:Single;var d8:Single):Integer;
VB	Function AC2_ReadUKT38sh4_trp (ByVal ch As Long, ByVal speed As Long, t1 As Single, t2 As Single, t3 As Single, t4 As Single, t5 As Single, t6 As Single, t7 As Single, t8 As Single, u1 As Single, u2 As Single, u3 As Single, u4 As Single, u5 As Single, u6 As Single, u7 As Single, u8 As Single, d1 As Single, d2 As Single, d3 As Single, d4 As Single, d5 As Single, d6 As Single, d7 As Single, d8 As Single) As Long

Параметры

ch

Номер линии АС2 (от 0 до 7).

speed

Скорость обмена, возможные значения см. в Руководстве по эксплуатации на соответствующее устройство.

t(X)

Возвращаемое значение температуры канала (X), или -300.0 при ошибках измерения.

u(X)

Возвращаемое значение уставки канала (X), или -300.0 при ошибках измерения.

d(X)

Возвращаемое значение дельты канала (X), или -300.0 при ошибках измерения.

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

5.2.8 AC2_ReadTRM32

Чтение данных из устройства ТРМ32.

C++	int AC2_ReadTRM32(DWORD ch,float &Taero, float &Tobr, float & Totop, float &Tgvs, float & TustObr, float & TustOtop,

	float &TustGvs, DWORD & UseDirectWater);
Delphi	function AC2_ReadTRM32(ch:DWORD;var Taero:Single;var Tobr:Single;var Totop:Single;var Tgvs:Single;var TustObr:Single;var TustOtop:Single;var TustGvs:Single;var UseDirectWater:DWORD):Integer;
VB	Function AC2_ReadTRM32 (ByVal ch As Long, Taero As Single, Tobr As Single, Totop As Single, Tgvs As Single, TustObr As Single, TustOtop As Single, TustGvs As Single, UseDirectWater As Long) As Long

Параметры

ch

Номер линии АС2 (от 0 до 7).

Taero

Температура наружного воздуха или температура прямой воды, зависит от режима работы.

Tobr

Температура обратной воды.

Totop

Температура воды в контуре отопления.

Tgvs

Температура воды в контуре горячего водоснабжения (ГВС).

TustObr

Расчетная уставка температуры обратной воды отопления.

TustOtop

Расчетная уставка температуры воды в контуре отопления.

TustGvs

Уставка температуры воды в контуре горячего водоснабжения.

UseDirectWater

Режим регулирования.

0000h	по температуре наружного воздуха
0001h	по температуре прямой воды

Возвращаемое значение

При успехе ERR_OK, при неудаче <0.

5.2.9 AC2_ReadTRM33

Чтение данных из устройства ТРМ33.

C++	int AC2_ReadTRM33(DWORD ch, float &t1, float &t2, float &t3, float &t4, float &t5, float &t6, float &t7, float &t8, float &Rshift_TobrMax, float &TavarMin, float &TustAero, float &TobrWaterLow);
Delphi	function AC2_ReadTRM33(ch:DWORD;var t1:Single;var t2:Single;var t3:Single;var t4:Single;var t5:Single;var t6:Single;var t7:Single;var t8:Single;var Rshift_TobrMax:Single;var TavarMin:Single;var TustAero:Single;var TobrWaterLow:Single):Integer;
VB	Function AC2_ReadTRM33 (ByVal ch As Long, t1 As Single, t2 As Single, t3 As Single, t4 As Single, t5 As Single, t6 As Single, t7 As Single, t8 As Single, Rshift_TobrMax As Single, TavarMin As Single, TustAero As Single, TobrWaterLow As Single) As Long

Параметры

ch

Номер линии АС2 (от 0 до 7).

t(X)

Возвращаемое значение температуры канала (X), или -300.0 при ошибках измерения.

Rshift_TobrMax

Числовое значение сдвига (Δ_{max}) относительно графика $T_{обр.гр.} = f(T_{наруж.})$ при определении $T_{обр.max}$.

TavarMin

Значение температуры приточного воздуха ($T_{авар.}$), ниже которой система переводится в режим защиты от замораживания

TustAero

Уставка температуры по приточному воздуху $T_{уст}$.

TobrWaterLow

Разрешенная величина снижения температуры обратной воды ниже отопительного графика.

Возвращаемое значение
При успехе ERR_OK, при неудаче <0.

6 Список кодов ошибок

6.1 Ошибки библиотеки

Ошибка	Описание
ERR_OK (0)	успешное завершение операции
ERR_INVALID_ARG (-1)	аргумент функции неверен
ERR_NO_RESOURCE (-2)	попытка использовать неинициализированный ресурс (например, неоткрытый порт)
ERR_RESOURCE_BUSY (-4)	ресурс уже используется (COM-порт уже используется другим потоком)
ERR_INVALID_RESOURCE (-5)	ошибка инициализации ресурса (например, при открытии COM-порта)
ERR_UNSUPPORTED (-6)	функция не поддерживается
ERR_BUFFER_TOO_SMALL (-7)	размер буфера меньше необходимого
ERR_BUFFER_INVALID (-8)	передан нулевой указатель на буфер
ERR_SDOT_VALUE (-9)	значение параметра слишком мало или слишком велико
ERR_ILLEGAL_PARAM_NAME (-10)	недопустимое имя параметра

6.2 Ошибки обмена

Ошибка	Описание
ERR_IO (-100)	общая ошибка обмена
ERR_FORMAT (-101)	неверный размер принятого кадра или неверный размер поля данных
ERR_TIMEOUT (-102)	прибор не ответил на запрос
ERR_CRC (-103)	неверная контрольная сумма в принятом от прибора кадре
ERR_NERR (-104)	прибор вернул код сетевой ошибки nErr
ERR_DEVERR (-105)	прибор вернул признак исключительной ситуации
ERR_INVALID_ANSWER (-106)	принят неполный кадр, кадр содержит недопустимые символы либо ошибочные данные

6.3 Исключительные ситуации

Код	Описание
0x0	значение заведомо неверно (вычисление невозможно)
0x1	попытка записать в параметр неверное значение
0x6	данные не готовы
0x7	датчик отключен

0x8	велика температура свободных концов ТП
0x9	мала температура свободных концов ТП
0xA	вычисленное значение слишком велико
0xB	вычисленное значение слишком мало
0xC	короткое замыкание
0xD	обрыв датчика
0xE	отсутствие связи с АЦП
0xF	некорректный калибровочный коэффициент
0x3000001	у регулятора отсутствует уставка (регулятор не работает в данный момент)
0x3000002	у регулятора уставка типа <мощность>
0x5000000	неверное количество входов
0x5000001	отключен вход вычислителя
0x5000002	отключен датчик, используемый вычислителем
0x5000003	датчик и вычислитель не соответствуют друг другу
0x5000004	отключен фильтр, используемый вычислителем
0x5000005	недопустимый формат данных фильтра RS-485
0x5000006	мала температура сухого термометра
0x5000007	велика температура сухого термометра
0x5000008	мала температура влажного термометра
0x5000009	велика температура влажного термометра
0x500000A	вычислитель отключен
0x500000B	на входе вычислителя корня отрицательное число
0x500000C	неверно задан индекс датчика
0x500000D	неверно задан индекс сетевого фильтра