

2017



БИБЛИОТЕКИ OWENIONET

РУКОВОДСТВО ПРОГРАММИСТА

Версия: 1.1
Дата: 04.12.2017

Содержание

1.	Принципы работы с библиотекой.....	3
1.1	Пример совместного использования.....	4
1.2	Конвертеры типов данных	5
2.	Описание библиотеки OwenioNet.....	5
2.1	Описание основных классов для передачи данных по протоколу Овен..	5
2.1.1	IOwenProtocolMaster	5
2.1.2	OwenProtocolMaster	6
2.1.3	OwenTransport	10
2.1.4	IStreamResource	11
2.1.5	SerialPortAdapter.....	12
2.1.6	enum AddressLengthType	16
2.1.7	IPackageLogger	16
2.1.8	SlaveNetworkException	17
3.	Описание библиотеки OwenioNet.DataConverter	20
3.1	Описание конвертеров данных.....	20
3.1.1	ConverterASCII	20
3.1.2	ConverterBool.....	21
3.1.3	ConverterClkFrmDateTime	22
3.1.4	ConverterClkFrmTime.....	24
3.1.5	ConverterDst.....	25
3.1.6	ConverterDecDotS.....	27
3.1.7	ConverterDecDotU	28
3.1.8	ConverterFloat	31
3.1.9	ConverterFloatTimestamp	32
3.1.10	ConverterI	34
3.1.11	ConverterParpoint.....	35
3.1.12	ConverterStoredDot	36
3.1.13	ConverterU.....	37
3.1.14	ConverterBaseDot<T>.....	39
3.2	Описание типов данных библиотеки	40

3.2.1	struct BoolValue	40
3.2.2	ClkFrmDateTime	40
3.2.3	struct Dst	41
3.2.4	FixedPoint	42
3.2.5	UnsignedFixedPoint.....	45
3.2.6	enum FormatStorageType.....	47
3.2.7	struct Parpoint	48
3.2.8	struct ValueWithTimestamp<T>	48
3.2.9	ValueConverter<T>	49
3.2.10	IValueConverter<T>.....	49
3.2.11	SlaveDataException.....	50
3.3	Реализация собственных конвертеров.....	51

1. Принципы работы с библиотекой

Для работы с устройством используйте библиотеки [OwenioNet](#) и [OwenioNet.DataConverter](#). [OwenioNet](#) отвечает за связь с устройством. [OwenioNet.DataConverter](#) отвечает за преобразования данных протокола Овен.

Для передачи данных используется интерфейс [IOwenProtocolMaster](#). Его реализует класс [OwenProtocolMaster](#). Класс [OwenProtocolMaster](#) имеет 2 фабричных метода, один использует [SerialPortAdapter](#), другой - произвольную реализацию интерфейса [IStreamResource](#).

Создание через объект [SerialPort](#):

```
using (var owenProtocol = OwenProtocolMaster
    .Create(new SerialPort("COM3", 9600, Parity.Even, 1, StopBits.One)))
{
```

Создание через объект реализующий интерфейс [IStreamResource](#):

```
using (var owenProtocol = OwenProtocolMaster
    .Create(new SerialPortAdapter(3, 9600, Parity.Even, 1, StopBits.One), null))
{
```

После того как объект [OwenProtocolMaster](#) создан, можно отправлять запросы для чтения и записи значений параметров.

Для записи значения параметра используйте метод [OwenWrite](#).

```
using (var owenProtocol = OwenProtocolMaster.Create(
    new SerialPortAdapter(3, 9600, Parity.Even, 1, StopBits.One), null))
{
    owenProtocol.OwenWrite(0x265, AddressLengthType.Bits11, "ab.L",
        new byte[] {0x45, 0x87}); //без индекса
    owenProtocol.OwenWrite(0x265, AddressLengthType.Bits11, "ab.L",
        new byte[] {0x45, 0x87}, 0x3456); //с индексом
}
```

Для чтения значения параметра используйте метод [OwenRead](#).

```
using (var owenProtocol = OwenProtocolMaster.Create(
    new SerialPortAdapter(3, 9600, Parity.Even, 1, StopBits.One), null))
{
    owenProtocol.OwenRead(0x265, AddressLengthType.Bits11, "ab.L"); //без индекса
    owenProtocol.OwenRead(0x265, AddressLengthType.Bits11, "ab.L", 0x3456); //с индексом
}
```

Для записи в лог принятых и отправленных сообщений необходимо реализовать интерфейс [IPackageLogger](#).

Значения, принимаемые и возвращаемые методами чтения и записи [IOwenProtocolMaster](#), представляют собой массивы байт. Для конвертирования

массивов байт в значения необходимого типа данных используйте классы `OwenioNet.DataConverter`.

Для конвертации значения в массив байт и обратно необходимо создать конвертер. Для примера рассмотрим [ConverterDecDotS](#). Сначала нужно создать объект конвертера (почти у всех конвертеров можно указывать, в каком выходном формате будут храниться данные). А затем вызвать метод [Convert](#) либо [ConvertBack](#) для конвертации.

```
ConverterDecDotS converter=new ConverterDecDotS();
byte[] convertedValue = converter.Convert(new FixedPoint(12520567, 3, true));
FixedPoint convertBack=converter.ConvertBack(convertedValue);
```

В примере выше показано, как объект [FixedPoint](#) конвертируется сначала в массив байт, а затем снова в объект `FixedPoint`.

Также некоторым конвертерам можно задавать размеры выходного массива. Для примера рассмотрим [ConverterFloat](#). Общий принцип работы схож с примером, рассмотренным выше.

```
ConverterFloat converterFloat=new ConverterFloat();
byte[] byteView=converterFloat.Convert((float) -8000.2);
```

1.1 Пример совместного использования

```
using (var owenProtocol = OwenProtocolMaster.Create(
    new SerialPortAdapter(3, 9600, Parity.Even, 1, StopBits.One), null))
{
    byte[] dataFromDevice = owenProtocol.OwenRead(0x265, AddressLengthType.Bits11,
        "ab.L");
    var converterFloat = new ConverterFloat();
    var valueFromDevice = converterFloat.ConvertBack(dataFromDevice);

    var valueToDevice = 10f;
    var dataToDevice = converterFloat.Convert(valueToDevice);
    owenProtocol.OwenWrite(0x265, AddressLengthType.Bits11, "ab.L", dataToDevice);
}
```

1.2 Конвертеры типов данных

Конвертер для типа данных	Класс-конвертер	Модификатор времени	Форма записи	Дополнительные параметры	Возвращаемый тип
DEC_dot (0-3)	ConverterDecDotU	без метки времени	двоичный, двоично-десятичная	Положение точки, Размер	UnsignedFixedPoint
DEC_dot_SGND (0-3)	ConverterDecDotS	без метки времени	двоичный, двоично-десятичная	Положение точки, Размер	FixedPoint
STORED_DOT	ConverterStoredDot	без метки времени	двоичный	Размер	FixedPoint
U	ConverterU	без метки времени	двоичный, двоично-десятичный	Размер	UInt32
Float	ConverterFloatTimesamp ConverterFloat	с меткой времени, без метки времени	двоичный	Размер	Single
DST	ConverterDst	без метки времени	двоичный	-	Структура DST
ASCII	ConverterAscii	без метки времени	двоичный	Длина строки	String (в приборе кодировка 1251)
I	ConverterI	без метки времени	двоичный, двоично-десятичный	Размер	Int32
CLK_frm_Time	ConverterClkFrmTime	без метки времени	двоичный, двоично-десятичный	-	TimeSpan
Parpoint	ConverterParpoint	без метки времени	двоичная	-	Структура Parpoint
Bool	ConverterBool	без метки времени	двоичная	Размер, значение 0 и 1	Bool
CLK_frm_DateTime	ConverterClkFrmDateTime	без метки времени	двоичный, двоично-десятичный	-	DateTime

2. Описание библиотеки OwenioNet

Библиотека предназначена для связи с устройствами, работающими по протоколу Овен.

2.1 Описание основных классов для передачи данных по протоколу Овен

2.1.1 IOwenProtocolMaster

Namespace: OwenioNet.

Интерфейс, обеспечивающий спецификацию для обмена данными по протоколу Овен.

Properties

Доступ	Тип	Наименование	Описание
get	IPackageLogger	PackageLogger	Объект, реализующий интерфейс IPackageLogger .
get set	bool	DisableLogging	Позволяет включать и отключать отладочный логгер пакетов.
get	OwenTransport	Transport	Получает объект OwenTransport , которому делегируется отправка сообщений.

2.1.2 OwenProtocolMaster

Namespace: OwenioNet.

Класс наследник [IOwenProtocolMaster](#). Инкапсулирует запись/чтение данных в объект, реализующий интерфейс [IStreamResource](#), с использованием методов Read и Write. Также позволяет логировать принятые/переданные пакеты с использованием класса, реализующего интерфейс [IPackageLogger](#). На основе переданных конструктору параметров, если необходимо, создаёт объект типа [OwenTrasport](#), которому и делегирует запись/чтение данных на транспортном уровне.

Create

Создаёт объект на основе переданных параметров (фабричный метод).

Access: static

Parameters:

Тип	Наименование	Описание
IStreamResource	streamResource	Объект, реализующий интерфейс IStreamResource ; необходим для создания OwenTransport . Осуществляет запись на транспортном уровне.
IPackageLogger	logging	Объект, реализующий интерфейс IPackageLogger для логирования пакетов при их записи/чтении. Не является обязательным.

Returns: OwenProtocolMaster

Throws:

Тип	Условие
System.ArgumentNullException	streamResource равен null.

Create

Создаёт объект на основе переданных параметров.

Access: static

Parameters:

Тип	Наименование	Описание
SerialPort	serialPort	Объект типа System.IO.Ports.SerialPort для записи данных в последовательный порт; используется для создания SerialPortAdapter , чтение/запись транспортного уровня.
IPackageLogger	logging	Объект, реализующий интерфейс IPackageLogger для логирования пакетов при их записи/чтении. Не является обязательным.

Returns: OwenProtocolMaster

Throws:

Тип	Условие
System.ArgumentNullException	serialPort равен null.

Properties

Доступ	Тип	Наименование	Описание
get	IPackageLogger	PackageLogger	Объект, реализующий интерфейс IPackageLogger .
get set	bool	DisableLogging	Позволяет включать и отключать отладочный логгер пакетов.
get	OwenTransport	Transport	Объект, осуществляющий чтение/запись данных на транспортном уровне.

OpenSerialPort

Создаёт адаптер для работы с последовательным портом ([SerialPortAdapter](#)), затем с помощью него создаёт новый объект типа [OwenTransport](#).

Parameters:

Тип	Наименование	Описание
<code>int</code>	<code>portNumber</code>	Номер порта (цифра после COM, например 1, 2, ... Чтобы получилось название порта).
<code>int</code>	<code>baudRate</code>	Скорость передачи в бодах.
<code>Parity</code>	<code>parity</code>	Протокол контроля чётности.
<code>int</code>	<code>dataBits</code>	Число битов данных.
<code>StopBits</code>	<code>stopBits</code>	Число стоповых битов в байте.
<code>IPackageLogger</code>	<code>logging</code>	Объект для логирования принятых/переданных пакетов.

Returns: void.

Throws:

Тип	Условие
<code>System.IO.IOException</code>	При неудачном создании SerialPort .

OwenWrite

На основе переданных параметров создаёт сообщение для [OwenTransport](#). Сообщение-ответ анализируется транспортным уровнем ([OwenTransport](#)), но не возвращается. Перехваченные исключения записываются в зависимости от значения свойства [DisableLogging](#).

Parameters:

Тип	Наименование	Описание
<code>int</code>	<code>address</code>	Адрес устройства, используются только последние 8 или 11 бит.
<code>AddressLengthType</code>	<code>addressLength</code>	Значение перечисления AddressLengthType ; в зависимости от этого параметра из адреса будут использоваться последние 8 или 11 бит.
<code>string</code>	<code>parameterName</code>	Строковое значение имени параметра.
<code>byte[]</code>	<code>dataField</code>	Поле данных. Не может быть null или не соответствовать описанию протокола, в противном случае будет вызвано исключение.
<code>ushort?</code>	<code>parameterIndex</code>	Значение последних 2 байт в блоке данных, которые отвечают за индекс параметра. Может быть null.

Returns: void.

Throws:

Тип	Условие
System.ArgumentNullException	Если dataField равен null.
System.FormatException	Если размер dataField не соответствует спецификации протокола Овен.

OwenRead

На основе переданных параметров создаёт сообщение для [OwenTransport](#). Полученный ответ анализируется на соответствие запросу и спецификации протокола.

Parameters:

Тип	Наименование	Описание
int	address	Адрес устройства, используются только последние 8 или 11 бит
AddressLengthType	addressLength	Значение перечисления AddressLengthType ; в зависимости от этого параметра из адреса будут использоваться последние 8 или 11 бит
string	parameterName	Строковое значение имени параметра
ushort?	parameterIndex	Значение последних 2 байт в блоке данных, которые отвечают за индекс параметра. Может быть null.

Returns: byte[] - блок данных (без индекса, если таковой имелся).

Throws:

Тип	Условие
System.Exception	Все исключения, которые возникают при передаче.

CloseSerialPort

Выполняет освобождение ресурсов, занимаемых объектом [OwenTransport](#) (см. свойство Transport), затем устанавливает свойство Transport в null.

Parameters: no.

Returns: void.

Throws: no.

Dispose

Выполняет освобождение ресурсов, занимаемых объектом [OwenTransport](#) (см. свойство Transport).

Parameters: no.

Returns: void.

Throws: no.

Dispose

Предотвращает повторное освобождение ресурсов, занятых объектом Transport.

Parameters:

Тип	Наименование	Описание
bool	disposing	Признак того, что нужно освободить ресурсы.

Returns: void.

Throws: no.

2.1.3 OwenTransport

Namespace: OwenioNet.IO.

Base: IDisposable.

Класс отвечает за приём и отправку кадра в формате протокола Овен, т.е. осуществляет преобразование данных в кадр транспортного уровня, а затем канального для записи в порт ввода-вывода.

OwenTransport

Properties

Доступ	Тип	Наименование	Описание
get	IPackageLogger	Logging	Получает и устанавливает объект, используемый для логирования принятых /переданных данных.
get set	int	Retries	Получает и устанавливает количество повторных попыток отправки данных в случае неудачи, значение по умолчанию – 3.
get set	int	ReadTimeout	Получает или задаёт timeout для чтения данных.
get set	int	WriteTimeout	Получает или задаёт timeout для записи данных.
get set	int	WaitToNextRequestMilliseconds	Получает или задаёт паузу между следующим ответом или запросом.

Dispose

Выполняет освобождение ресурсов, которые занимает объект, реализующий [IStreamResource](#) (см. свойство StreamResource).

Parameters: no.

Returns: void.

Throws: no.

Dispose

Предотвращает повторное освобождение ресурсов, занятых объектом `IStreamResource`.

Parameters:

Тип	Наименование	Описание
<code>bool</code>	<code>disposing</code>	Признак того, что нужно освободить ресурсы.

Returns: void.

Throws: no.

2.1.4 IStreamResource

Namespace: `OwenioNet.IO`.

Base: `IDisposable`.

Интерфейс, обеспечивающий спецификацию для работы с портом ввода-вывода. Если необходимо добавить в библиотеку новый тип объекта, куда будет осуществляться ввод/вывод данных по протоколу Овен, то класс-адаптер должен реализовывать данный интерфейс. После этого библиотека сможет работать с этим устройством (см. [SerialPortAdapter](#)).

Open

Открывает порт ввода-вывода.

Parameters: no.

Returns: void.

Close

Закрывает порт ввода-вывода.

Parameters: no.

Returns: void.

Read

Считывает массив байт из порта на основе переданных параметров.

Parameters:

Тип	Наименование	Описание
byte[]	buffer	Массив, в который считываются данные.
int	offset	Смещение от начала; с этого индекса будут считываться данные.
int	count	Количество байт, которое необходимо считать.

Returns: int. Количество считанных байт из потока.

Write

Записывает массив байт в порт ввода-вывода.

Parameters:

Тип	Наименование	Описание
byte[]	buffer	Массив байт, который будет записан в порт.
int	offset	Смещение в буфере.
int	count	Количество байт, которые необходимо записать в порт.

Returns: void.

DiscardInBuffer

Очищает входной буфер.

Parameters: no.

Returns: void.

Properties

Доступ	Тип	Наименование	Описание
get	bool	IsOpened	Возвращает признак того, что порт открыт.
get set	int	ReadTimeout	Устанавливает или получает значение timeout в миллисекундах при чтении данных из порта.
get set	int	WriteTimeout	Устанавливает или получает значение timeout в миллисекундах при записи данных в порт.

2.1.5 SerialPortAdapter

Namespace: OwenioNet.IO.

Base: IStreamResource.

Класс адаптер, для удобного включения [System.IO.Port.SerialPort](#) в библиотеку Owenio.Net. Транслирует вызовы через интерфейс к объекту SerialPort.

SerialPortAdapter

Создаёт объект SerialPortAdapter, параметры используются для создания объекта SerialPort.

Parameters:

Тип	Наименование	Описание
int	portNumber	Номер порта (цифра после COM, например 1, 2, ... Чтобы получилось название порта).
int	baudRate	Скорость передачи в бодах.
Parity	parity	Протокол контроля чётности.
int	dataBits	Число битов данных.
StopBits	stopBits	Число стоповых битов в байте.

Throws:

Тип	Условие
System.IO.IOException	См SerialPort.(String, Int32, Parity, Int32, StopBits)

SerialPortAdapter

Создаёт объект SerialPortAdapter.

Parameters:

Тип	Наименование	Описание
SerialPort	serialPort	Объект, который используется для ввода/вывода информации.

Throws:

Тип	Условие
System.NullReferenceException	serialPort равен null

Open

Открывает порт ввода-вывода.

Parameters: no.

Returns: void.

Throws:

Тип	Условие
Все исключения, которые возникают в SerialPort.Open	См SerialPort.Open()

Close

Закрывает порт ввода-вывода.

Parameters: no.

Returns: void.

Throws:

Тип	Условие
Все исключения, которые возникают в SerialPort.Close	См SerialPort.Close()

Read

Считывает массив байт из порта на основе переданных параметров.

Parameters:

Тип	Наименование	Описание
byte[]	buffer	Массив, в который считываются данные.
int	offset	Смещение от начала; с этого индекса будут считываться данные.
int	count	Количество байт, которое необходимо считать.

Returns: int. Количество считанных байт из потока.

Throws:

Тип	Условие
Все исключения, которые возникают в SerialPort.Read	См. SerialPort.Read

Write

Записывает массив байт в порт ввода-вывода.

Parameters:

Тип	Наименование	Описание
<code>byte[]</code>	<code>buffer</code>	Массив байт, который будет записан в порт.
<code>int</code>	<code>offset</code>	Смещение в буфере.
<code>int</code>	<code>count</code>	Количество байт, которые необходимо записать в порт.

Returns: void.

Throws:

Тип	Условие
Все исключения, которые возникают в <code>SerialPort.Write</code>	См SerialPort.Write

DiscardInBuffer

Очищает входной буфер.

Parameters: no.

Returns: void.

Throws:

Тип	Условие
Все исключения, которые возникают в <code>SerialPort.DiscardInBuffer</code>	См. SerialPort.DiscardInBuffer

Dispose

Выполняет освобождение ресурсов, занимаемых объектом.

Parameters: no.

Returns: void.

Throws: no.

Dispose

Предотвращает повторное освобождение ресурсов, занятых объектом.

Parameters:

Тип	Наименование	Описание
<code>bool</code>	<code>disposing</code>	Признак того, что нужно освободить ресурсы.

Returns: void.

Throws: no.

Properties

Реализация интерфейса IStreamResource.

Доступ	Тип	Наименование	Описание
get	bool	IsOpened	Возвращает признак того, что порт открыт.
get set	int	ReadTimeout	Устанавливает или получает значение timeout в миллисекундах при чтении данных из порта.
get set	int	WriteTimeout	Устанавливает или получает значение timeout в миллисекундах при записи данных в порт.

2.1.6 enum AddressLengthType

Namespace: OwenioNet.Types.

Type: int.

Перечисление возможных вариантов длины адреса устройства в соответствии со спецификацией протокола.

Значения:

Наименование	Описание
Bits8	8 битовый адрес устройства.
Bits11	11 битовый адрес устройства.

2.1.7 IPackageLogger

Namespace: OwenioNet.Log.

Интерфейс-адаптер, обеспечивающий спецификацию для записи запросов по протоколу Овен.

LogWritePackage

Записывает запрос на изменение значения параметра в журнал.

Parameters:

Тип	Наименование	Описание
byte[]	package	Массив байт, который будет записан в журнал.

Returns: void.

LogWritePackage

Записывает запрос на изменение значения параметра в журнал.

Parameters:

Тип	Наименование	Описание
<code>string</code>	message	Информационное сообщение.
<code>byte[]</code>	package	Массив байт, который будет записан в журнал.

Returns: void.

LogWritePackage

Записывает запрос на изменение значения параметра в журнал.

Parameters:

Тип	Наименование	Описание
<code>string</code>	message	Информационное сообщение.

Returns: void.

LogReadPackage

Записывает запрос на чтение значения параметра в журнал.

Parameters:

Тип	Наименование	Описание
<code>string</code>	message	Информационное сообщение.

Returns: void.

LogReadPackage

Записывает запрос на изменение значения параметра в журнал.

Parameters:

Тип	Наименование	Описание
<code>string</code>	message	Информационное сообщение.
<code>byte[]</code>	package	Массив байт, который будет записан в журнал.

Returns: void.

LogReadPackage

Записывает запрос на изменение значения параметра в журнал.

Parameters:

Тип	Наименование	Описание
<code>string</code>	message	Информационное сообщение.

Returns: void.

2.1.8 SlaveNetworkException

Base: System.Exception.

Namespace: OwenioNet.Exceptions.

Данный класс предназначен для хранения и передачи информации об исключении на более высокие уровни приложения.

SlaveNetworkException

Заполняет все значения полей класса значениями по умолчанию.

Parameters: no.

Throws: no.

SlaveNetworkException

Parameters:

Тип	Наименование	Описание
<code>string</code>	message	Сообщение об ошибке, передаётся в базовый класс как аргумент конструктора.

Throws: no.

SlaveNetworkException

Parameters:

Тип	Наименование	Описание
<code>string</code>	message	Сообщение об ошибке, передаётся в базовый класс как аргумент конструктора.
<code>Exception</code>	innerException	Внутреннее исключение.

Throws: no.

SlaveNetworkException

Parameters:

Тип	Наименование	Описание
<code>SlaveNetworkExceptionResponse</code>	slaveNetworkExceptionResponse	Объект, инкапсулирующий информацию о сетевой ошибке. Используется при создании строкового представления объекта.
<code>string</code>	message	Сообщение об ошибке (передаётся в базовый класс).

Throws:

Тип	Условие
<code>System.ArgumentNullException</code>	Параметр <code>message</code> равен null.

Properties

Доступ	Тип	Наименование	Описание
get	string	Message	Получает текст сообщения об ошибке. Конкатенация строкового представления SlaveNetworkExceptionResponse и свойства Message базового класса.
get	byte	NetworkErrorCode	Получает код сетевой ошибки из объекта SlaveNetworkExceptionResponse, если объект отсутствует, то 0.
get	ushort	ParameterHash	Получает хэш представление параметра из объекта SlaveNetworkExceptionResponse, если объект отсутствует, то 0.
get	int	Address	Получает адрес устройства, при обмене с которым произошла исключительная ситуация (из объекта SlaveNetworkExceptionResponse). Если объект или адрес устройства отсутствуют, то 0.

3. Описание библиотеки OwenioNet.DataConverter

Данная библиотека представляет собой уровень приложения для работы с протоколом Овен. При работе с протоколом с помощью данной библиотеки можно осуществлять конвертацию данных как для передачи (в массив байт), так и после получения данных (массив байт в выходной формат конвертера).

3.1 Описание конвертеров данных

3.1.1 ConverterASCII

Base: ValueConverter<string>

Namespace: OwenioNet.DataConverter.Converter.

Позволяет конвертировать строку с кодировкой Win1251 в массив байт в прямом и обратном направлениях. Максимальная длина строки – 15 символов.

ConverterASCII

Устанавливает значение свойства Length равным значению параметра.

Parameters:

Тип	Наименование	Описание
int	length	Длина строки (см. описание метода Convert ниже).

Throws:

Тип	Условие
System.ArgumentException	Недопустимая длина строки [1, DataConverter.MaxDataSize].

Convert

Выполняет конвертацию строки в массив байт. Если длина строки меньше значения свойства Length, то строка дополняется пробелами справа. Если длина строки больше значения свойства Length, то строка усекается справа.

Parameters:

Тип	Наименование	Описание
string	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentNullException	Параметр равен null.

ConvertBack

Выполняет конвертацию массива байт в строку.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: string. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	Длина массива байт больше значения свойства Length.

Properties

Доступ	Тип	Наименование	Описание
get	bool	Length	Длина строки в символах/размер массива в байтах. Интерпретация значения зависит от конвертации. Значение по умолчанию - 1.

3.1.2 ConverterBool

Base: ValueConverter<bool>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует значение типа bool в массив байт и обратно.

ConverterBool

Устанавливает значения свойств Value и Size переданными параметрами.

Parameters:

Тип	Наименование	Описание
boolValue	value	Эталонные значения признаков истины и лжи.
int	size	Размер массива в байтах. Значение по умолчанию - 2.

Throws:

Тип	Условие
System.ArgumentException	Значение size меньше 1 или больше 2.

Convert

Выполняет конвертацию значения в массив байт. Если аргумент - true, то будет конвертироваться значение Value.TrueValue. Иначе будет конвертироваться значение Value.FalseValue.

Parameters:

Тип	Наименование	Описание
bool	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws: no.

ConvertBack

Выполняет конвертацию массива байт в булево значение.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: bool. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	Длина массива байт не соответствует значению свойства Size.

Properties

Доступ	Тип	Наименование	Описание
get	int	Size	Размер переменной в блоке данных.
get	BoolValue	Value	Эталонные отрицательное и положительное значения.

3.1.3 ConverterClkFrmDateTime

Base: ConverterBaseClkFrm<DateTime>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует DateTime (только компоненту даты) в массив байт и обратно. По спецификации часового формата данных протокола Овен (см. описание протокола пункт 5.1.3).

ConverterClkFrmDateTime

Вызывает конструктор базового класса с переданными параметрами.

Parameters:

Тип	Наименование	Описание
byte	template	Самый последний байт блока данных, должен быть заполнен в соответствии с п. 5.1.3 описания протокола.
int	size	Размер блока данных в байтах.
FormatStorageType	format	Формат хранения данных (двоичный или двоично-десятичный).

Throws:

Тип	Условие
System.ArgumentException	Значение size меньше 1 или больше 15.

Convert

Выполняет конвертацию значения в массив байт.

Parameters:

Тип	Наименование	Описание
DateTime	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentOutOfRangeException	Значение цены блока в шаблоне не соответствует спецификации.

ConvertBack

Выполняет конвертацию массива байт в DateTime. Работает только с компонентами даты.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: DateTime. Преобразованное значение.

Throws:

Тип	Условие
System.FormatException	Значение цены блока в шаблоне не соответствует спецификации или неверный размер блока данных в байтах.

3.1.4 ConverterClkFrmTime

Base: ConverterBaseClkFrm<TimeSpan>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует TimeSpan в массив байт и обратно. По спецификации часового формата данных протокола Овен (см. описание протокола пункт 5.1.3).

ConverterClkFrmTime

Вызывает конструктор базового класса с переданными параметрами.

Parameters:

Тип	Наименование	Описание
byte	template	Самый последний байт блока данных, должен быть заполнен в соответствии с п. 5.1.3 описания протокола.
int	size	Размер блока данных в байтах.
FormatStorageType	format	Формат хранения данных (двоичный или двоично-десятичный).

Throws:

Тип	Условие
System.ArgumentException	Значение size меньше 1 или больше 15.

Convert

Выполняет конвертацию значения в массив байт.

Parameters:

Тип	Наименование	Описание
TimeSpan	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws:

Тип	Условие
System.FormatException	Неверный шаблон часового формата.

ConvertBack

Выполняет конвертацию массива байт в DateTime. Работает только с компонентами даты.

Parameters:

Тип	Наименование	Описание
<code>byte[]</code>	<code>dataField</code>	Значение для конвертации.

Returns: TimeSpan. Преобразованное значение.

Throws:

Тип	Условие
System.FormatException	Значение цены блока в шаблоне не соответствует спецификации или неверный размер блока данных в байтах.

3.1.5 ConverterDst

Base: ValueConverter<Dst>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует значение типа Dst в массив байт и обратно.

ConverterDecDotS

Инициализирует значение свойства Size.

Parameters:

Тип	Наименование	Описание
<code>int</code>	<code>size</code>	Размер массива в байтах при обратной конвертации.

Throws:

Тип	Условие
System.ArgumentException	Значение <code>size</code> меньше 1 или больше 2.

Convert

Выполняет преобразование типа Dst в массив байт. Значение дней умножается на 100, затем к нему прибавляется значение месяцев. Полученное число приводится к типу ushort и преобразуется в массив байт.

Parameters:

Тип	Наименование	Описание
<code>Dst</code>	<code>value</code>	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	Количество месяцев не принадлежит отрезку [1, 12] или количество дней не принадлежит отрезку [1, 31].

ConvertBack

Выполняет преобразование массива байт в тип Dst, аналогично методу Convert, но в обратном порядке.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: Dst. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	<ul style="list-style-type: none">• Количество месяцев не принадлежит отрезку [1, 12];• количество дней не принадлежит отрезку [1, 31];• длина входного массива в байтах не соответствует значению свойства Size.

DataFieldValidate

Используется для проверки массива байт перед обратной конвертацией.

Access: protected.

Parameters:

Тип	Наименование	Описание
<code>byte[]</code>	<code>dataField</code>	Блок данных, который будет участвовать в обратной конвертации.
<code>string</code>	<code>typeName</code>	Название типа данных, в который будет происходить преобразование.

Returns: void.

Throws:

Тип	Условие
<code>System.ArgumentNullException</code>	Параметр <code>dataField</code> равен null.
<code>System.ArgumentException</code>	Параметр <code>dataField</code> имеет недопустимый размер.
<code>SlaveDataException</code>	При конвертации типов <code>uint</code> , <code>int</code> , <code>float</code> , <code>double</code> – старший полубайт младшего байта содержит признак ошибки.

Свойства

Доступ	Тип	Наименование	Описание
<code>get</code>	<code>int</code>	<code>Size</code>	Размер массива байт при обратной конвертации.

3.1.6 ConverterDecDotS

Base: ConverterBaseDot<FixedPoint>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует значение типа `FixedPoint` в поле данных в соответствии с описанием протокола Овен.

ConverterDecDotS

Вызывает конструктор базового класса с переданным параметром.

Parameters:

Тип	Наименование	Описание
<code>FormatStorageType</code>	<code>format</code>	Формат хранения данных (двоичный или двоично-десятичный).

Throws: no.

Convert

Вызывает метод базового класса для выполнения преобразования.

Parameters:

Тип	Наименование	Описание
<code>FixedPoint</code>	<code>value</code>	Значение для конвертации.

Returns: `byte[]`. Преобразованное значение.

Throws:

Тип	Условие
<code>System.ArgumentException</code>	Количество значимых знаков после запятой больше 3.
<code>System.ArgumentNullException</code>	Параметр равен null.

ConvertBack

Вызывает метод базового класса для конвертации массива байт в `FixedPoint`.

Parameters:

Тип	Наименование	Описание
<code>byte[]</code>	<code>dataField</code>	Значение для конвертации.

Returns: `FixedPoint`. Преобразованное значение.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
<code>get</code>	<code>FormatStorageType</code>	<code>Format</code>	Формат хранения данных.

3.1.7 ConverterDecDotU

Base: `ConverterBaseDot<UnsignedFixedPoint>`

Namespace: `OwenioNet.DataConverter.Converter`

Конвертирует значение типа `UnsignedFixedPoint` в поле данных в соответствии с описанием протокола Овен.

ConverterDecDotS

Вызывает конструктор базового класса с переданным параметром.

Parameters:

Тип	Наименование	Описание
<code>FormatStorageType</code>	<code>format</code>	Формат хранения данных (двоичный или двоично-десятичный).

Throws: no.

Convert

Вызывает метод базового класса для выполнения преобразования.

Parameters:

Тип	Наименование	Описание
UnsignedFixedPoint	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	Количество значимых знаков после запятой больше 3.
System.ArgumentNullException	Параметр равен null.

ConvertBack

Вызывает метод базового класса для конвертации массива байт в [UnsignedFixedPoint](#).

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: UnsignedFixedPoint. Преобразованное значение.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
get	FormatStorageType	Format	Формат хранения данных.

Base: ValueConverter<Dst>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует значение типа Dst в массив байт и обратно.

ConverterDecDotS

Инициализирует значение свойства Size.

Parameters:

Тип	Наименование	Описание
<code>int</code>	<code>size</code>	Размер массива в байтах при обратной конвертации.

Throws:

Тип	Условие
<code>System.ArgumentException</code>	Значение <code>size</code> меньше 1 или больше 2.

Convert

Выполняет преобразование типа `Dst` в массив байт. Значение дней умножается на 100, затем к нему прибавляется значение месяцев. Полученное число приводится к типу `ushort` и преобразуется в массив байт.

Parameters:

Тип	Наименование	Описание
<code>Dst</code>	<code>value</code>	Значение для конвертации.

Returns: `byte[]`. Преобразованное значение.

Throws:

Тип	Условие
<code>System.ArgumentException</code>	Количество месяцев не принадлежит отрезку [1, 12] или количество дней не принадлежит отрезку [1, 31].

ConvertBack

Выполняет преобразование массива байт в тип `Dst`, аналогично методу `Convert`, но в обратном порядке.

Parameters:

Тип	Наименование	Описание
<code>byte[]</code>	<code>dataField</code>	Значение для конвертации.

Returns: `Dst`. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	<ul style="list-style-type: none">• Количество месяцев не принадлежит отрезку [1, 12];• количество дней не принадлежит отрезку [1, 31];• длина входного массива в байтах не соответствует значению свойства Size.

Properties

Доступ	Тип	Наименование	Описание
get	int	Size	Размер массива байт при обратной конвертации.

3.1.8 ConverterFloat

Base: ValueConverter<float>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует значение типа float в массив байт и обратно. Допустимы 4-байтовый float и 3-байтовый. Подробнее см. описание протокола Овен.

ConverterFloat

Инициализирует значение свойства Size.

Parameters:

Тип	Наименование	Описание
int	size	Размер массива в байтах.

Правила прямой и обратной конвертации зависят от значения параметра int:

- 1 – в блоке данных содержится код ошибки. При таком значении параметра конвертер должен использоваться только для обработки полученных с прибора данных;
- 3 – конвертирование по правилам pic float;
- 4 – конвертирование по правилам IEEE 754.

Throws:

Тип	Условие
System.ArgumentException	Значение size не равно 1,3,4.

Convert

Выполняет преобразование значения float в массив байт. Размер выходного массива в байтах будет равен значению свойства Size.

Parameters:

Тип	Наименование	Описание
float	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws: no.

ConvertBack

Метод конвертирует массивы из 3 и 4 байтов в значение float. Если размер массива - 1 байт, то генерируется исключение [SlaveDataException](#).

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: float. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	Размер массива dataField не соответствует значению свойства Size.
SlaveDataException	Если размер массива dataField равен 1

Properties

Доступ	Тип	Наименование	Описание
get	int	Size	Размер массива байт при обратной конвертации.

3.1.9 ConverterFloatTimestamp

Base: ValueWithTimestamp<ConverterFloat, float>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует значение типа float и временную метку ([TimeSpan](#)) в массив байт и обратно.

ConverterWithTimestamp

Вызывает конструктор базового класса с объектом типа [ConverterFloat](#), конструктору которого передаётся аргумент size.

Parameters:

Тип	Наименование	Описание
<code>int</code>	<code>size</code>	Размер массива в байтах при обратной конвертации.

Throws: no.

Convert

Convert(TV value, TimeSpan timestamp=null) из базового класса.

Выполняет преобразование аргументов в массив байт.

Parameters:

Тип	Наименование	Описание
<code>TV</code>	<code>value</code>	Шаблонный тип для конвертации.
<code>TimeSpan?</code>	<code>timestamp</code>	Значение TimeSpan, имеет значение по умолчанию – null.

Returns: byte[]. Преобразованное значение.

Throws:

Тип	Условие
<code>System.FormatException</code>	Размер массива байт после конвертации значений превышает максимально допустимый.

ConvertBack

Преобразует массив байт в значение ValueWithTimestamp<float>.

Parameters:

Тип	Наименование	Описание
<code>byte[]</code>	<code>dataField</code>	Значение для конвертации.

Returns: ValueWithTimestamp<float>. Преобразованное значение.

Throws:

Тип	Условие
<code>System.ArgumentNullException</code>	Параметр dataField равен null.

Properties

Доступ	Тип	Наименование	Описание
<code>get</code>	<code>TimeSpan?</code>	<code>timestamp</code>	Значение TimeSpan, которое будет преобразовываться.

3.1.10 ConverterI

Base: ValueConverter<int>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует значение типа int со знаком в массив байт и обратно.

ConverterI

Инициализирует значения свойств Size и Format соответствующими параметрами конструктора.

Parameters:

Тип	Наименование	Описание
int	size	Размер массива в байтах. При прямой и обратной конвертации происходит проверка размера массива.
FormatStorageType	format	Формат представления данных после прямой конвертации.

Throws:

Тип	Условие
System.ArgumentException	Значение size меньше 1 или больше 4.

Convert

Конвертирует значение в массив байт.

Parameters:

Тип	Наименование	Описание
int	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	Попытка конвертации отрицательного числа в формат bcd.

ConvertBack

Конвертирует массив байт в значение int.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: int. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	При двоичном формате хранения. Размер входного массива не соответствует значению свойства Size.

Properties

Доступ	Тип	Наименование	Описание
get	int	Size	Размер переменной в блоке данных.
get	FormatStorageType	Format	Формат хранения данных.

3.1.11 ConverterParpoint

Base: ValueConverter<Parpoint>

Namespace: OwenioNet.DataConverter.Converter.

Access: public.

Конвертирует значение типа Parpoint в массив байт и обратно. Точность каждого числа – 1 знак после запятой.

ConverterParpoint

Инициализирует значение свойства Size значением переданного параметра.

Access: public.

Parameters:

Тип	Наименование	Описание
int	size	Размер массива в байтах. При прямой и обратной конвертации происходит проверка размера массива.

Throws:

Тип	Условие
System.ArgumentException	Значение size не равно 4.

Convert

Конвертирует значение в массив байт.

Parameters:

Тип	Наименование	Описание
Parpoint	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws: no.

ConvertBack

Конвертирует массив байт в значение int.

Access: public.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: Parpoint. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentException	Размер входного массива не соответствует значению свойства Size.

Properties

Доступ	Тип	Наименование	Описание
get	FormatStorageType	Format	Формат хранения данных.

3.1.12 ConverterStoredDot

Base: ConverterBaseDot<FixedPoint>

Namespace: OwenioNet.DataConverter.Converter.

Конвертирует значение типа FixedPoint в массив байт и обратно.

ConverterStoredDot

Вызывает конструктор базового класса с параметром.

Parameters:

Тип	Наименование	Описание
FormatStorageType	format	Формат представления данных при конвертации.

Throws: no.

Convert

Конвертирует значение в массив байт. Вызывает метод базового класса для преобразования.

Parameters:

Тип	Наименование	Описание
<code>FixedPoint</code>	<code>value</code>	Значение для конвертации.

Returns: `byte[]`. Преобразованное значение.

Throws:

Тип	Условие
<code>System.ArgumentNullException</code>	Параметр <code>value</code> равен <code>null</code> .
<code>System.ArgumentException</code>	Значение <code>value.Exponent</code> больше 7.

ConvertBack

Конвертирует массив байт в значение `FixedPoint`. Вызывает метод базового класса для преобразования.

Parameters:

Тип	Наименование	Описание
<code>byte[]</code>	<code>dataField</code>	Значение для конвертации.

Returns: `FixedPoint`. Преобразованное значение.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
<code>get</code>	<code>FormatStorageType</code>	<code>Format</code>	Формат хранения данных.

3.1.13 ConverterU

Base: `ValueConverter<uint>`

Namespace: `OwenioNet.DataConverter.Converter`.

Конвертирует значение типа `uint` в массив байт и обратно.

ConverterU

Инициализирует значения свойств `Size` и `Format` соответствующими значениями аргументов.

Parameters:

Тип	Наименование	Описание
<code>int</code>	<code>size</code>	Размер массива в байтах. Допустимые значения 1, 2, 3, 4. Значение по умолчанию - 4.
<code>FormatStorageType</code>	<code>format</code>	Формат хранения данных после прямой конвертации.

Throws:

Тип	Условие
<code>System.ArgumentException</code>	Параметр <code>size</code> имеет недопустимое значение.

Convert

Конвертирует значение `uint` в массив байт.

Parameters:

Тип	Наименование	Описание
<code>uint</code>	<code>value</code>	Значение для конвертации.

Returns: `byte[]`. Преобразованное значение.

Throws: no.

ConvertBack

Конвертирует массив байт в значение `uint`.

Parameters:

Тип	Наименование	Описание
<code>byte[]</code>	<code>dataField</code>	Значение для конвертации.

Returns: `uint`. Преобразованное значение.

Throws:

Тип	Условие
<code>System.ArgumentException</code>	При двоичном формате хранения данных размер входного массива не соответствует значению свойства <code>Size</code> .

Properties

Доступ	Тип	Наименование	Описание
get	int	Size	Размер переменной в блоке данных.
get	FormatStorageType	Format	Формат хранения данных.

3.1.14 ConverterBaseDot<T>

Base: ValueConverter <T>

Namespace: OwenioNet.DataConverter.Converter.

Базовый класс для конвертирования наследников UnsignedFixedPoint. T может быть только UnsignedFixedPoint или классом наследником.

Convert

Конвертирует значение в массив байт. Вызывает метод базового класса для преобразования.

Parameters:

Тип	Наименование	Описание
T	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

Throws:

Тип	Условие
System.ArgumentNullException	Параметр value равен null.

ConvertBack

Конвертирует массив байт в значение FixedPoint. Вызывает метод базового класса для преобразования.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: T. Преобразованное значение.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
get	FormatStorageType	Format	Формат хранения данных.

3.2 Описание типов данных библиотеки

3.2.1 struct BoolValue

Namespace: OwenioNet.DataConverter.Types.

Позволяет конвертировать строку с кодировкой Win1251 в массив байт в прямом и обратном направлении. Максимальная длина строки – 15 символов.

BoolValue

Инициализирует свойства FalseValue и TrueValue. В конвертерах значение, преобразованное из массива, проверяется на равенство одному из значений, и на основе этого возвращается результат. При прямой конвертации в зависимости от параметра конвертера будет преобразовываться TrueValue либо FalseValue.

Access: public.

Parameters:

Тип	Наименование	Описание
ushort	falseValue	Значение, которое является признаком лжи.
ushort	trueValue	Значение, которое является признаком истины.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
get	ushort	FalseValue	Значение, которое является признаком лжи.
get	ushort	TrueValue	Значение, которое является признаком истины.

3.2.2 ClkFrmDateTime

Namespace: OwenioNet.DataConverter.Types.

Свойства описывают весь набор значений из 5.1.3 описания протокола. Используется внутри конвертеров, у которых предком является [ConverterBaseClkFrm](#).

Properties

Доступ	Тип	Наименование	Описание
get set	int	MSec	Миллисекунды
get set	int	Sec	Секунды
get set	int	Min	Минуты
get set	int	Hour	Часы
get set	int	Day	Дни
get set	int	Month	Месяцы
get set	int	Year	Годы

3.2.3 struct Dst

Namespace: OwenioNet.DataConverter.Types.

Представляет структуру для хранения количества дней и месяцев.

Dst

Инициализирует свойства Day и Month.

Parameters:

Тип	Наименование	Описание
byte	day	Количество дней: [1;31].
byte	month	Количество месяцев: [1;12].

Throws:

Тип	Условие
System.ArgumentOutOfRangeException	Значение дней[1'31] или месяцев[1'12] не соответствует допустимому диапазону.

Equals

Проверяет равенство количества дней и месяцев обоих экземпляров.

Parameters:

Тип	Наименование	Описание
object	obj	Второе значение для сравнения.

Returns: bool. Признак сравнения.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
get	byte	Day	Количество дней.
get	byte	Month	Количество месяцев.

3.2.4 FixedPoint

Namespace: OwenioNet.DataConverter.Types.

Представляет структуру для хранения числа со знаком соответствующего п. 5.1.2 описания протокола Овен.

FixedPoint

Вызывает конструктор базового класса с переданными параметрами.

Parameters:

Тип	Наименование	Описание
ulong	mantissa	Дробное десятичное число, преобразованное в целое с учётом значимых знаков после запятой.
byte	exponent	Количество значимых знаков после запятой[1;7].

Throws:

Тип	Условие
System.ArgumentException	Недопустимое значение параметра exponent.

FixedPoint

Вызывает конструктор базового класса с переданными параметрами (mantissa, exponent). Инициализирует свойство Sign.

Parameters:

Тип	Наименование	Описание
<code>ulong</code>	<code>mantissa</code>	Дробное десятичное число, преобразованное в целое с учётом значимых знаков после запятой.
<code>byte</code>	<code>exponent</code>	Количество значимых знаков после запятой[1;7].
<code>bool</code>	<code>sign</code>	Признак того, что число отрицательное.

Throws:

Тип	Условие
<code>System.ArgumentException</code>	Недопустимое значение параметра <code>exponent</code> .

FixedPoint

Вызывает конструктор базового класса. В качестве параметров передаются: абсолютное значение параметра `value`, 0, признак того, что `value` - отрицательное значение.

Parameters:

Тип	Наименование	Описание
<code>int</code>	<code>value</code>	Значение, из которого будет создан объект <code>FixedPoint</code> .

Throws:

Тип	Условие
<code>System.ArgumentException</code>	Недопустимое значение параметра <code>exponent</code> .

FixedPoint

Вызывает конструктор базового класса. Из параметров `value` и `exponent` высчитывается мантисса. Она передаётся в перегруженный конструктор данного класса вместе с `exponent` и признаком знака числа. Устанавливает значение свойства `Sign`.

Parameters:

Тип	Наименование	Описание
<code>double</code>	<code>value</code>	Значение, из которого будет создан объект <code>FixedPoint</code> .
<code>byte</code>	<code>exponent</code>	Количество значимых знаков после запятой.

Throws:

Тип	Условие
System.ArgumentException	Недопустимое значение параметра exponent.

FixedPoint

Вызывает конструктор базового класса, в качестве аргумента передаёт ему абсолютное значение параметра.

Parameters:

Тип	Наименование	Описание
double	value	Значение, из которого будет создан объект FixedPoint.

Throws: no.**FixedPoint**

Вызывает конструктор базового класса. Числа с плавающей точкой (double/float) при конвертации будут обработаны автоматически в зависимости от количества знаков после запятой и значения параметра digit:

- количество знаков превышает digit - то лишние знаки будут отрезаны,
- количество знаков меньше digit - то пустые регистры будут заполнены нулями.

Parameters:

Тип	Наименование	Описание
double	value	Значение, из которого будет создан объект FixedPoint.
byte	digit	Минимальное количество значащих цифр после запятой.

Throws: no.**AsDouble**

Представляет текущий объект как значение типа double.

Parameters: no.**Returns:** double. Значение объекта FixedPoint в виде double.**Throws:** no.**Operator double**

Оператор явного приведения объекта к типу double.

Access: static explicit.

Parameters: no.

Returns: double. Значение объекта FixedPoint в виде double.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
get	bool	Sign	Получает признак того, что конвертируемое число является отрицательным.

3.2.5 UnsignedFixedPoint

Namespace: OwenioNet.DataConverter.Types.

Access: public.

Представляет структуру для хранения положительного числа, соответствующего п. 5.1.2 описания протокола Овен.

UnsignedFixedPoint

Устанавливает значения свойств Mantissa, Exponent значениями параметров mantissa и exponent соответственно.

Parameters:

Тип	Наименование	Описание
ulong	mantissa	Дробное десятичное число, преобразованное в целое с учётом значимых цифр после запятой.
byte	exponent	Количество значимых цифр после запятой[1;7].

Throws:

Тип	Условие
System.ArgumentException	Недопустимое значение параметра exponent.

UnsignedFixedPoint

Вызывает перегрузку конструктора данного класса, предварительно вычислив мантиссу на основе параметров value и exponent.

Parameters:

Тип	Наименование	Описание
<code>double</code>	<code>value</code>	Число для преобразования.
<code>byte</code>	<code>exponent</code>	Количество значимых знаков после запятой[1;7].

Throws:

Тип	Условие
<code>System.ArgumentException</code>	Значение value меньше нуля.

UnsignedFixedPoint

Вызывает перегрузку конструктора данного класса, с переданным значением в качестве мантиссы и без значащих цифр после запятой.

Parameters:

Тип	Наименование	Описание
<code>uint</code>	<code>value</code>	Число для преобразования.

Throws: no.

UnsignedFixedPoint

Устанавливает значения свойств `Mantissa` и `Exponent` при помощи автоматического подбора точности.

Parameters:

Тип	Наименование	Описание
<code>double</code>	<code>value</code>	Число для преобразования.

Throws:

Тип	Условие
<code>System.ArgumentException</code>	Значение value меньше нуля.

UnsignedFixedPoint

Вызывает перегруженный конструктор класса, предварительно получив мантиссу из параметра `value` с учётом параметра `digit`. Значение мантиссы обрабатывается автоматически в зависимости от количества знаков после запятой и значения параметра `digit`:

- Если количество знаков превышает `digit` - то лишние знаки будут отрезаны,
- Если количество знаков меньше `digit` - то пустые регистры будут заполнены нулями.

Parameters:

Тип	Наименование	Описание
double	value	Число для преобразования.
byte	digit	Минимальное число знаков после запятой.

Throws:

Тип	Условие
System.ArgumentException	Значение value меньше нуля.

AsDouble

Представляет объект в виде double значения.

Parameters: no.

Returns: double. Представление в виде double.

Throws: no.

Operator double

Оператор явного приведения объекта к типу double.

Access: static explicit.

Parameters: no.

Returns: double. Значение объекта UnsignedFixedPoint в виде double.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
get	byte	Exponent	Позиция точки – количество значимых цифр после запятой.
get	ulong	Mantissa	Мантисса числа.

3.2.6 enum FormatStorageType

Namespace: OwenioNet.DataConverter.Types.

Формат хранения числа в виде байт.

Values:

Тип	Наименование	Описание
<code>int</code>	Bin	Двоичный формат.
<code>int</code>	Bcd	Двоично-десятичный формат.

3.2.7 struct Parpoint

Namespace: OwenioNet.DataConverter.Types.

Позволяет сохранять позицию точки. Используется в конвертере [ConverterParpoint](#).

Parpoint

Инициализирует свойства X и Y передаваемыми параметрами.

Parameters:

Тип	Наименование	Описание
<code>double</code>	x	Значение X.
<code>double</code>	y	Значение Y.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
<code>get</code>	<code>double</code>	x	Значение X.
<code>get</code>	<code>double</code>	y	Значение Y.

3.2.8 struct ValueWithTimestamp<T>

ValueWithTimestamp<T>

Namespace: OwenioNet.DataConverter.Types.

Позволяет сохранять значение произвольного типа и метку времени (TimeSpan).

Parpoint

Инициализирует свойства Value и Timestamp передаваемыми параметрами (value и timestamp).

Parameters:

Тип	Наименование	Описание
Template type T	value	Значение, к которому относится временная метка.
TimeSpan	timestamp	Временная метка.

Throws: no.

Properties

Доступ	Тип	Наименование	Описание
get	Template type T	Value	Значение, к которому относится временная метка.
get	TimeSpan	Timestamp	Временная метка.

3.2.9 ValueConverter<T>

Base: IValueConverter<T>

Namespace: OwenioNet.DataConverter.

Абстрактный класс, который определяет метод для проверки массива байт при обратной конвертации.

Convert

Конвертирует массив байт в значение. Используется при обратной конвертации.

Parameters:

Тип	Наименование	Описание
Template type T	value	Значение для конвертации.

Returns: byte[]. Преобразованное значение.

ConvertBack

Конвертирует массив байт в значение. Используется при обратной конвертации.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Значение для конвертации.

Returns: template type T. Преобразованное значение.

3.2.10 IValueConverter<T>

Namespace: OwenioNet.DataConverter.

Интерфейс, обеспечивающий спецификацию для работы с конвертерами данных.

Convert

Используется для прямой конвертации.

Parameters:

Тип	Наименование	Описание
Template type T	value	Значение для преобразования в массив байт.

Returns: byte[]. Преобразованное значение.

ConvertBack

Используется для обратной конвертации.

Parameters:

Тип	Наименование	Описание
byte[]	dataField	Массив байт для обратного преобразования в значение.

Returns: template type T. Преобразованное значение.

Размер массива byteView будет равен 3 – будет отброшен младший байт мантиссы.

3.2.11 SlaveDataException

Base: System.Exception.

Namespace: OwenioNet.Exceptions.

Данный класс предназначен для хранения и передачи информации об исключении на более высокие уровни приложения.

Properties

Доступ	Тип	Наименование	Описание
get	string	Message	Получает текст сообщения об ошибке. Текстовое описание кода ошибки данных протокола Овен.
get	int	ExceptionCode	Код ошибки по протоколу Овен.

3.3 Реализация собственных конвертеров

Если возникла необходимость реализовать собственный конвертер, то его лучше наследовать от класса [ValueConverter](#). Он обеспечит общий интерфейс для конвертации, а также метод для проверки входного массива при обратной конвертации.

Если нужно реализовать дополнительный тип данных, который будет храниться вместе с временной меткой, то он должен использовать структуру [ValueWithTimestamp](#)<T>.

При написании конвертера для работы с целыми числами или числами с фиксированной точкой для хранения таких чисел лучше всего создать класс наследник `FixedPoint` или `UnsignedFixedPoint`, а сам конвертер унаследовать от [ConverterBaseDot](#). Методы этого класса позволят преобразовывать значения, производные от классов `(Unsigned)FixedPoint`.