

# Библиотека OWEN

## Реализация протокола ОВЕН на языке программирования Tcl

### Руководство пользователя

18 апреля 2013 г.

## Содержание

<b>1</b>	<b>Общие сведения</b>	<b>1</b>
<b>2</b>	<b>Описание библиотеки</b>	<b>2</b>
2.1	Дескриптор . . . . .	2
2.2	Использование порта . . . . .	2
2.3	Информация об ошибке . . . . .	2
<b>3</b>	<b>Описание команд</b>	<b>4</b>
3.1	::owen::configure . . . . .	4
3.2	::owen::lastStatus . . . . .	5
3.3	::owen::lastError . . . . .	6
3.4	::owen::sendCommand . . . . .	6
3.5	::owen::readString . . . . .	7
3.6	::owen::readInt . . . . .	8
3.7	::owen::writeInt8 . . . . .	8
3.8	::owen::writeInt16 . . . . .	9
3.9	::owen::readFloat24 . . . . .	10
3.10	::owen::writeFloat24 . . . . .	11

## 1 Общие сведения

Данная библиотека реализует протокол обмена компании ОВЕН на языке программирования Tcl. Данный протокол используется для взаимо-

действия с устройствами производства данной компании через последовательный порт. Описание протокола можно найти на сайте компании [owen.ru](http://owen.ru).

В данной версии библиотека позволяет:

- отправлять команды на устройство;
- считывать строковые параметры из устройства;
- считывать и записывать целочисленные параметры;
- считывать и записывать параметры в формате FLOAT24 (24-битовое число с плавающей точкой).

## 2 Описание библиотеки

### 2.1 Дескриптор

Перед началом обмена с устройством необходимо создать *дескриптор* — структура, описывающая параметры устройства, такие как адрес последовательного порта и его установки, адрес самого устройства, величина таймаута и пр. Дескриптор создаётся командой `::owen::configure`:

```
# Последовательный порт COM1, адрес устройства 32
set desc [::owen::configure -port COM1 -addr 32]
# Дескриптор создан
# Теперь запрашиваем параметр DEV
set dev [::owen::readString $desc DEV]
```

Полученный дескриптор далее используется для обмена.

### 2.2 Использование порта

Последовательный порт, по которому происходит обмен, открывается непосредственно перед передачей данных и закрывается сразу после получения ответа от устройства.

### 2.3 Информация об ошибке

Все команды обмена с устройством, которые возвращают какое-либо значение, в случае ошибки возвращают пустую строку без генерации исключительной ситуации. Чтобы узнать детальную информацию об ошибке, используются команды `:owen::lastStatus` и `::owen::lastError`.

Команда `::owen::lastStatus` возвращает *статус* последней выполненной операции. Если статус равен `::owen::STATUS_OK`, операция прошла успешно и нет необходимости вызывать команду `::owen::lastError`.

- `::owen::STATUS_OK` — операция прошла успешно, не нужно далее вызывать `::owen::lastError`.
- `::owen::STATUS_EXCEPTION` — устройство сигнализирует об исключительной ситуации, нужно вызвать команду `::owen::lastError`, чтобы получить код ситуации. Например, код `0xFE` говорит об отсутствии связи с АЦП. Расшифровка кодов исключительных ситуаций см. в документации к устройству.
- `::owen::STATUS_NETWORK_ERROR` — произошла ошибка обмена данными. Как правило, данный статус говорит о попытке получить или записать значение параметра с неверным типом значения. Например, целочисленный параметр считывается командой `::owen::readFloat24`, или 16-битовое целочисленное значение записывается командой `::owen::writeInt8`. В этом случае необходимо вызвать команду `::owen::lastError`, чтобы узнать код ошибки. Список кодов ошибок приведён в документации к устройству. Например, код `0x31` соответствует ошибке «Размер поля данных не соответствует ожидаемому». Также имеется несколько дополнительных кодов ошибок:
  - `::owen::ERROR_BAD_DATA` — полученные данные не являются корректным пакетом. Вероятно требуется уменьшить скорость обмена.
  - `::owen::ERROR_BAD_LENGTH` — полученный пакет имеет неверную длину. Нужно проверить правильность работы устройства.
  - `::owen::ERROR_TIMEOUT` — таймаут ожидания ответа от устройства. Необходимо проверить правильность физического подключения и питание устройства.
- `::owen::STATUS_PORT_ERROR` — произошла ошибка открытия или конфигурирования последовательного порта. Команда `::owen::lastError` вернёт текстовое описание ошибки. Например, данная ситуация может произойти при попытке соединения с устройством по отсутствующему порту.

Команда `::owen::lastError` возвращает детальное описание ошибки. Трактовки значения зависит от статуса, см. выше. В зависимости от

статуса команда возвращает либо числовой код, либо текстовое описание ошибки.

Информация об ошибке сохраняется в глобальной переменной, поэтому если производится работа с несколькими устройствами одновременно, нужно сохранять статус и ошибку в локальных переменных сразу после операции над каждым устройством.

## 3 Описание команд

### 3.1 `::owen::configure`

#### Назначение

Создаёт дескриптор устройства по заданным параметрам. Не производит открытие или проверку существования последовательного порта.

#### Синтакс

```
::owen::configure args
```

#### Аргументы

- **args** — Параметры устройства в виде пар «параметр»-«значение». Список возможных параметров:
  - **-port** — имя последовательного порта, по которому происходит обмен данными, например, «COM1».
  - **-settings** — параметры последовательного порта в виде строки, например, «9600,8,n,1».
  - **-timeout** — таймаут ожидания ответа от устройства в мс, например, «500».
  - **-numOfAttempts** — число попыток связи с устройством в случае невозможности открыть порт или помех на линии, например, «3».
  - **-addr** — адрес устройства, например, «16».
  - **-addrType** — длина адреса: 0 — 8 бит, 1 — 11 бит. Можно использовать predefined константы `::owen::ADDR_TYPE_8BIT` и `::owen::ADDR_TYPE_11BIT`.

### Возвращаемое значение

Дескриптор устройства.

### Пример

```
# устройство подключено к порту COM1
# и 8-битный адрес 32
set desc [::owen::configure -port COM1 -addr 32 -addrType 0]
# в переменной desc - дескриптор
```

## 3.2 ::owen::lastStatus

### Назначение

Возвращает статус последней произведённой операции. Обычно используется в паре с `::owen::lastError`

### Синтакс

```
::owen::lastStatus
```

### Аргументы

Отсутствуют

### Возвращаемое значение

Целочисленное число с кодом статуса.

### Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32 -addrType 0]
# считываем параметр DEV
set dev [::owen::readString $desc DEV]
# проверяем статус
if { [::owen::lastStatus] != $::owen::STATUS_OK } {
    error "Error reading DEV parameter"
}
```

### См. также

```
::owen::lastError
```

### 3.3 `::owen::lastError`

#### Назначение

Возвращает описание ошибки от последней произведённой операции. Обычно используется в паре с `::owen::lastStatus`

#### Синтакс

```
::owen::lastError
```

#### Аргументы

Отсутствуют

#### Возвращаемое значение

Целочисленное число с кодом или текстовое описание ошибки. Трактовка кода ошибки зависит от статуса.

#### Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32 -addrType 0]
# считываем параметр DEV
set dev [::owen::readString $desc DEV]
# проверяем статус
if { [::owen::lastStatus] != $::owen::STATUS_OK } {
    error "Error [::lastError]"
}
```

#### См. также

```
::owen::lastStatus
```

### 3.4 `::owen::sendCommand`

#### Назначение

Отправляет команду устройству без ожидания ответа.

#### Синтакс

```
::owen::sendCommand desc cmd
```

## Аргументы

- `desc` — дескриптор, полученный командой `::owen::configure`.
- `cmd` — строка с командой.

## Возвращаемое значение

Отсутствует

## Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32]
# перезагрузим устройство
::owen::sendCommand $desc INIT
```

## 3.5 ::owen::readString

### Назначение

Считывает строковый параметр из устройства.

### Синтакс

```
::owen::readString desc parameter
```

## Аргументы

- `desc` — дескриптор, полученный командой `::owen::configure`.
- `parameter` — имя параметра.

## Возвращаемое значение

Значение параметра или пустая строка в случае ошибки.

## Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32]
# считывает тип устройства
set dev [::owen::readString $desc DEV]
```

### 3.6 `::owen::readInt`

#### Назначение

Считывает целочисленный параметр из устройства.

#### Синтакс

```
::owen::readInt desc parameter ?index?
```

#### Аргументы

- `desc` — дескриптор, полученный командой `::owen::configure`.
- `parameter` — имя параметра.
- `index` — индекс параметра. Может отсутствовать, если параметр не имеет индекса.

#### Возвращаемое значение

Значение параметра или пустая строка в случае ошибки.

#### Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32]
# считываем адрес устройства
set addr [::owen::readInt $desc ADDR]
```

### 3.7 `::owen::writeInt8`

#### Назначение

Записывает целочисленный 8-битный параметр в устройство, после чего считывает и возвращает этот параметр.

#### Синтакс

```
::owen::writeInt8 desc parameter index value
```



## Аргументы

- `desc` — дескриптор, полученный командой `::owen::configure`.
- `parameter` — имя параметра.
- `index` — индекс параметра. Если параметр не имеет индекса, нужно указать -1.
- `value` — Значение параметра.

## Возвращаемое значение

Значение параметра или пустая строка в случае ошибки.

## Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32]
# время задержки
set rsdl [::owen::writeInt8 $desc RSDL -1 20]
```

## 3.8 ::owen::writeInt16

### Назначение

Записывает целочисленный 16-битный параметр в устройство, после чего считывает и возвращает этот параметр.

### Синтакс

```
::owen::writeInt8 desc parameter index value
```

## Аргументы

- `desc` — дескриптор, полученный командой `::owen::configure`.
- `parameter` — имя параметра.
- `index` — индекс параметра. Если параметр не имеет индекса, нужно указать -1.
- `value` — Значение параметра.

### Возвращаемое значение

Значение параметра или пустая строка в случае ошибки.

### Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32]
# задаём новый адрес
set addr [::owen::writeInt16 $desc ADDR -1 16]
```

## 3.9 ::owen::readFloat24

### Назначение

Считывает числовой параметр в формате FLOAT24 из устройства.

### Синтакс

```
::owen::readFloat24 desc parameter ?index?
```

### Аргументы

- `desc` — дескриптор, полученный командой `::owen::configure`.
- `parameter` — имя параметра.
- `index` — индекс параметра. Может отсутствовать, если параметр не имеет индекса.

### Возвращаемое значение

Значение параметра или пустая строка в случае ошибки.

### Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32]
# считываем уставку
set setPoint [::owen::readFloat24 $desc SP 0]
```

### 3.10 `::owen::writeFloat24`

#### Назначение

Записывает числовой параметр в формате FLOAT24 в устройство, после чего считывает и возвращает этот параметр.

#### Синтакс

```
::owen::writeFloat24 desc parameter index value
```

#### Аргументы

- `desc` — дескриптор, полученный командой `::owen::configure`.
- `parameter` — имя параметра.
- `index` — индекс параметра. Если параметр не имеет индекса, нужно указать -1.
- `value` — Значение параметра.

#### Возвращаемое значение

Значение параметра или пустая строка в случае ошибки.

#### Пример

```
# создаём дескриптор
set desc [::owen::configure -port COM1 -addr 32]
# новая уставка
set setPoint [::owen::writeFloat24 $desc SP 0 123.4]
```