

Классы TransportModbus, ModbusRegister



Содержание

ModbusRegister

Описание

Поля

Методы

ModbusCounter

Описание

Перечисления

Свойства

Перечисления

TransportModbusRtu

Описание

Свойства

Методы

Обработчики событий

1 ModbusRegister

1.1 Описание

Класс, представляющий регистр Modbus.

Предок: BasicSource.

1.2 Поля

```
| uint8   m_ucDeviceNo      ///< Номер устройства.
| uint8   m_ucReadFunctionNo ///< Номер функции чтения.
| uint16  m_wReadRegisterNo ///< Номер регистра чтения.
| uint8   m_ucWriteFunctionNo ///< Номер функции записи.
| uint16  m_wWriteRegisterNo ///< Номер регистра записи.
| uint8   m_ucSize          ///< Размер регистра в битах.
| uint8   m_ucPriority       ///< Приоритет. Не используется.
```

1.3 Методы

```
| int ModbusRegister::WriteValue(SignalValue sv) ///  
| в устройство.
```

2 ModbusCounter

2.1 Описание

Класс, представляющий служебную переменную-счетчик, хранящую статистику обмена с подчиненными устройствами и признак наличия соединения.

Предок: [BasicSource](#).

2.2 Перечисления

```
| enum ModbusCounter::EnCounterType  
| {  
|     ctUndefined,    // Не определено, ошибка (не используется)  
|     ctSend,         // Отправлено пакетов  
|     ctReceive,      // Принято пакетов  
|     ctError,        // Количество ошибок (CRC, пропуски ответов)  
|     ctConnection    // Признак наличия соединения (1/0)  
| }  
|
```

2.3 Свойства

```
| uint8 DeviceNo      // Номер устройства  
|  
| EnCounterType CounterType // Тип счетчика
```

3 Перечисления

EnParityType — Режим проверки четности:

- ptUndefined — Не определено.
- ptNone — Без проверки четности.
- ptOdd — Проверка на нечетность.
- ptEven — Проверка на четность.

EnStopbits — Количество стоповых бит:

- sbUndefined — Не определено.
- sbOne — 1 стоповый бит.
- sbOnepointfive — 1.5 стоповых бита.
- sbTwo — 2 стоповых бита.

EnTransportMode — Режим работы транспорта Modbus:

- tmUndefined — Не определено.
- tmMaster — Режим ведущего (Master).
- tmSlave — Режим ведомого (Slave).

EnET2Action — Действие при достижении порога ошибок второго уровня (ErrorsThreshold2):

- taUndefined — Не определено.
- taNothing — Действие отсутствует (ничего не предпринимать).
- taReconnect — Переподключение транспорта.

4 TransportModbusRtu

4.1 Описание

Класс, реализующий транспорт Modbus-RTU для передачи данных через последовательный интерфейс (COM-порт).

Предок: [BasicNode](#).

4.2 Свойства

Свойство	Тип	Описание
Mode	EnTransportMode	Режим работы транспорта.
ResponseTimeoutMs	int	Тайм-аут ожидания ответа от устройства (в мс).
PacketDelay	int	Задержка между пакетами.
AllowRequestGaps	bool	Разрешение на наличие пауз в запросах.
Connected	bool	Признак наличия соединения (только чтение).
Running	bool	Состояние работы транспорта. При true изменение списка регистров/счетчиков запрещено.
ErrorsThreshold1	int	Порог ошибок первого уровня.
ErrorsThreshold2	int	Порог ошибок второго уровня.
ET2Action	EnET2Action	Действие при достижении порога ошибок 2-го уровня.

Свойство	Тип	Описание
Baudrate	int	Скорость передачи данных (бит/с).
Port	string	Имя последовательного порта (например, "COM1").
Parity	EnParityType	Режим проверки четности.
Stopbits	EnStopbits	Количество стоповых бит.

4.3 Методы

```
void TransportModbusRTU::Start() // Запуск работы транспорта. Менять набор регистров можно только на
остановленном транспорте (см. свойство Running).

void TransportModbusRTU::Stop() // Остановка работы транспорта. Менять набор регистров можно только на
```

```

| остановленном транспорте (см. свойство Running).
|
| -----
| bool TransportModbusRTU::AddRegister(ModbusRegister@ reg) // Добавление регистра в список опроса.
| Возвращает true, если операция успешна (транспорт остановлен).
|
| -----
| bool TransportModbusRTU::RemoveRegister(ModbusRegister@ reg) // Удаление регистра из списка опроса.
| Возвращает true, если операция успешна.
|
| -----
| bool TransportModbusRTU::RemoveRegisters() // Удаление всех регистров.
|
| -----
| array<ModbusRegister@> TransportModbusRTU::GetRegisters() // Получение текущего списка добавленных
| регистров.
|
| -----
| bool TransportModbusRTU::AddCounter(ModbusCounter@ cnt) // Добавление счетчика статистики.
|
| -----
| bool TransportModbusRTU::RemoveCounter(ModbusCounter@ cnt) // Удаление счетчика статистики.
|
| -----
| bool TransportModbusRTU::RemoveCounters() // Удаление всех счетчиков.
|
| -----
| array<ModbusCounter@> TransportModbusRTU::GetCounters() // Получение текущего списка добавленных
| счетчиков.
|
| -----

```

4.4 Обработчики событий

- OnSourceUpdated — Событие, возникающее при обновлении значения регистра (чтение).
- OnSourceWriteStart — Событие, возникающее при начале процесса записи значения в регистр.

Источник — https://docs.kb-agava.ru/index.php?title=Классы_TransportModbus,_ModbusRegister&oldid=3577

Эта страница в последний раз была отредактирована 15 июля 2026 в 12:28.