

ModbusRegister

Поля

Методы

BasicWidget

Методы

Обработчики

BasicWindow

Методы

1 BasicNode

Базовый класс для всех узлов проекта.

Предок: нет.

1.1 Конструкторы

```
| BasicNode::BasicNode()           ///< Конструктор по умолчанию  
| BasicNode::BasicNode(string path) ///< Создание узла по пути в проекте
```

1.2 Методы

```
| string BasicNode::GetName()           ///< Получение имени узла  
| string BasicNode::GetShortName()      ///< Получение шифра узла  
| string BasicNode::GetDescription()    ///< Получение описания узла  
|  
| void BasicNode::SetName(string name)  
| void BasicNode::SetShortName(string shortname)  
| void BasicNode::SetDescription(string description)  
|  
| string BasicNode::GetPath()           ///< Получение пути узла в проекте  
|  
| void BasicNode::Start()               ///< Запуск узла  
| void BasicNode::Stop()                ///< Остановка узла  
|  
| NodeProperty@ BasicNode::GetProperty(string& prop) ///< Получение указателя на свойство узла  
| void BasicNode::PropertyValueUpdated(string& prop) ///< Сигнализация об обновлении значения  
| свойства
```

1.3 Свойства

Имя	Описание	Тип	Доступ	Внутренний идентификатор
Id	Идентификатор узла	int	Чтение/Запись	CONFIGNODE_PARAM_ID
Name	Имя узла	string	Чтение/Запись	CONFIGNODE_PARAM_NAME
Description	Описание узла	string	Чтение/Запись	CONFIGNODE_PARAM_DESCRIPTION
ShortName	Шифр узла	string	Чтение/Запись	CONFIGNODE_PARAM_SHORTNAME
NodeDisabled	Узел выключен	bool	Чтение/Запись	CONFIGNODE_PARAM_DISABLED

2 SignalValue

Класс, хранящий значение, передаваемое между узлами.

Предок: нет.

2.1 Конструкторы

```
| SignalValue::SignalValue(string value)
| SignalValue::SignalValue(int value)
| SignalValue::SignalValue(double value)
| SignalValue::SignalValue(bool value)
```

2.2 Методы

```
| string SignalValue::ToString()    ///< Преобразовать значение в строку
| int SignalValue::ToInt()         ///< Преобразовать значение в int
| double SignalValue::ToDouble()   ///< Преобразовать значение в double
| bool SignalValue::ToBool()       ///< Преобразовать значение в bool
```

```
| void SignalValue::SetValue(string& value)
| void SignalValue::SetValue(int value)
| void SignalValue::SetValue(double value)
| void SignalValue::SetValue(bool value)
```

```
| bool SignalValue::IsNumber()     ///< Значение является числовым
| bool SignalValue::IsError()      ///< Значение является ошибкой
```

2.3 Примеры

```
| void fnIncrementSource(BasicSource_t &source)
| {
|     SignalValue sv = source.GetValue();
|     if (sv.IsNumber() == true)
|     {
|         int iVal = sv.ToInt();
|         iVal++;
|
|         sv = SignalValue(iVal);
|     }
|     else
|     {
|         sv = SignalValue(0);
|     }
|     source.SetValue(sv);
| }
```

3 NodeProperty

Класс, предоставляющий доступ к свойствам узлов.

Предок: нет.

3.1 Конструкторы

```
| NodeProperty::NodeProperty()  
| NodeProperty::NodeProperty(NodeProperty& other)
```

3.2 Методы

```
| string NodeProperty::GetName()           ///  
| string NodeProperty::GetShortName()     ///  
| string NodeProperty::GetDescription()   ///  
| string NodeProperty::GetTitle()        ///  
|
```

```
| string NodeProperty::ToString()         ///  
| int NodeProperty::ToInt()              ///  
| double NodeProperty::ToDouble()        ///  
| bool NodeProperty::ToBool()            ///  
|
```

```
| int NodeProperty::SetValue(string val)  ///  
| int NodeProperty::SetValue(int val)     ///  
| int NodeProperty::SetValue(double val)  ///  
| int NodeProperty::SetValue(bool val)   ///  
|
```

3.3 Примеры

```
| BasicNode node("/Root/node");  
| NodeProperty@ prop = node.GetProperty("ShortName");  
|  
| string sn2 = prop.ToString();  
| prop.SetValue(string(sn2+"ShortName1"));
```

4 BasicSource

Предок: [BasicNode](#).

4.1 Методы

```
| SignalValue BasicSource::GetValue()     ///  
|
```

Метод производит получение значения источника.

Результат - значение типа [SignalValue](#).

```
| int BasicSource::SetValue(SignalValue sv) ///  
| значений связанных узлов (в транспорте).
```

Метод производит установку нового значения источника. Если источник связан с транспортом, то транспорт выполнит передачу нового значения в связанное с ним внешнее устройство.

Результат - код выполнения операции.

```
| int BasicSource::UpdateValue(SignalValue sv) ///< Установка собственного значения.
```

Метод производит установку нового значения источника без передачи значения транспорту.

Результат - код выполнения операции.

5 BasicSignal

Предок: BasicSource.

5.1 Методы

```
| SignalValue BasicSignal::GetValue()          ///< Получение значения.
```

Метод производит получение значения сигнала.

Результат - значение типа SignalValue.

```
| int BasicSignal::SetValue(SignalValue sv)    ///< Установка собственного значения с установкой значений связанных узлов (источников).
```

Метод производит установку нового значения сигнала. Если сигнал связан с источником (или несколькими источниками), то будет выполнена передача нового значения в источник(-и).

Результат - код выполнения операции.

```
| int BasicSignal::UpdateValue(SignalValue sv) ///< Установка собственного значения.
```

Метод производит установку нового значения источника без передачи значения источнику.

Результат - код выполнения операции.

6 ModbusRegister

Предок: BasicSource.

6.1 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.
```

Источник — https://docs.kb-agava.ru/index.php?title=Базовые_классы_AgavaSCADA/AgavaPLC&oldid=3476

Эта страница в последний раз была отредактирована 7 апреля 2026 в 08:54.