PlcBitAddress Members

Namespace: IPS7Lnk.Advanced
Assemblies: IPS7LnkNet.Advanced.dll, IPS7LnkNet.Advanced.dll
The PlcBitAddress type exposes the following members.

Constructors

PlcBitAddress(PlcOperand, Int32, Int32)

Initializes a new instance of the PlcBitAddress class using the specified `operand`, `byteNumber` and `bitNumber`.

C#

```
public PlcBitAddress(PlcOperand operand, int byteNumber, int bitNumber)
```

Parameters

`operand` PlcOperand

The PlcOperand to that the new PlcBitAddress refers.

`byteNumber` Int32

The number of the byte to that the new PlcBitAddress refers.

`bitNumber` Int32

The number of the bit to that the new PlcBitAddress refers.

Exceptions

ArgumentException

The type of `operand` specified is not supported or does not support a bit number.

ArgumentNullException

The `operand` is a null reference (Nothing in Visual Basic).

ArgumentOutOfRangeException

The `byteNumber` or `bitNumber` is out of the bounds defined by MinByteNumber, MaxByteNumber, MinBitNumber or MaxBitNumber.

PlcBitAddress(PlcOperandType, Int32, Int32)

Initializes a new instance of the PlcBitAddress class using the specified `operandType`, `byteNumber` and `bitNumber`.

C#

```
public PlcBitAddress(PlcOperandType operandType, int byteNumber, int bitNumber)
```

Parameters

`operandType` PlcOperandType

The PlcOperandType to that the new PlcBitAddress refers.

`byteNumber` Int32

The number of the byte to that the new PlcBitAddress refers.

`bitNumber` Int32

The number of the bit to that the new PlcBitAddress refers.

Exceptions

ArgumentException

The `operandType` specified requires an operand number or does not support a bit number.

ArgumentOutOfRangeException

The `byteNumber` or `bitNumber` is out of the bounds defined by MinByteNumber, MaxByteNumber, MinBitNumber or MaxBitNumber.

PlcBitAddress(PlcOperandType, Int32, Int32, Int32)

Initializes a new instance of the PlcBitAddress class using the specified `operandType`, `operandNumber`, `byteNumber` and `bitNumber`.

C#

```
public PlcBitAddress(PlcOperandType operandType, int operandNumber, int byteNumber, int
bitNumber)
```

Parameters

`operandType` PlcOperandType

The PlcOperandType to that the new PlcBitAddress refers.

`operandNumber` Int32

The number of the operand to that the new PlcBitAddress refers.

`byteNumber` Int32

The number of the byte to that the new PlcBitAddress refers.

`bitNumber` Int32

The number of the bit to that the new PlcBitAddress refers.

Exceptions

ArgumentException

The `operandType` specified is not supported the `operandType` specified does not support an operand number.

ArgumentOutOfRangeException

The `operandNumber`, `byteNumber` or `bitNumber` is out of the bounds defined by MinOperandNumber, MaxOperandNumber, MinByteNumber, MaxByteNumber, MinBitNumber or MaxBitNumber.

PlcBitAddress(SerializationInfo, StreamingContext)

Initializes a new instance of the PlcBitAddress class with serialized data.

C#

```
protected PlcBitAddress(SerializationInfo info, StreamingContext context)
```

Parameters

`info` Serialization.SerializationInfo

The Serialization.SerializationInfo that holds the serialized object data about the exception being thrown.

`context` Serialization.StreamingContext

The Serialization.StreamingContext that contains contextual information about the source or destination.

Exceptions

ArgumentNullException

The info parameter is null reference (Nothing in Visual Basic).

Serialization.SerializationException

The class name is null reference (Nothing in Visual Basic) or HResult is zero (0).

Methods

CompareTo(Object)

Compares the current PlcBitAddress with the `other`.

C#

```
public override int CompareTo(object other)
```

Parameters

`other` Object

The PlcBitAddress to compare with this PlcBitAddress.

Returns

Int32

A 32-bit signed integer that indicates the relative order of the objects being compared (CompareTo(Object)).

CompareTo(PlcBitAddress)

Compares the current PlcBitAddress with another PlcBitAddress.

C#

```
public int CompareTo(PlcBitAddress other)
```

Parameters

`other` PlcBitAddress

The PlcBitAddress to compare with this PlcBitAddress.

Returns

Int32

A 32-bit signed integer that indicates the relative order of the objects being compared (CompareTo()).

Equals(Object)

Determines whether the specified `other` is equal to this PlcBitAddress.

C#

```
public override bool Equals(object other)
```

Parameters

`other` Object

The PlcBitAddress to compare to the current PlcBitAddress.

Returns

Boolean

The value true if the specified PlcBitAddress is equal to the current PlcBitAddress; otherwise the value false.

Equals(PlcBitAddress)

Determines whether the specified `other` is equal to this PlcBitAddress.

C#

```
public bool Equals(PlcBitAddress other)
```

Parameters

`other` PlcBitAddress

The PlcBitAddress to compare to the current PlcBitAddress.

Returns

Boolean

The value true if the specified PlcBitAddress is equal to the current PlcBitAddress; otherwise the value false.

GetHashCode()

Retrieves a hash code for this PlcBitAddress.

C#

```
public override int GetHashCode()
```

Returns

Int32

An Int32 that contains the hash code for the PlcBitAddress.

Operators

Equality(PlcBitAddress, PlcBitAddress)

Returns a value indicating whether two instance of PlcBitAddress are equal.

C#

```
public static bool operator ==(PlcBitAddress left, PlcBitAddress right)
```

GreaterThan(PlcBitAddress, PlcBitAddress)

Determines whether the first specified PlcBitAddress object is greater than the second specified PlcBitAddress object.

C#

```
public static bool operator>(PlcBitAddress left, PlcBitAddress right)
```

GreaterThanOrEqual(PlcBitAddress, PlcBitAddress)

Determines whether the first specified PlcBitAddress object is greater than or equal to the second specified PlcBitAddress object.

C#

```
public static bool operator >=(PlcBitAddress left, PlcBitAddress right)
```

Implicit(String to PlcBitAddress)

Converts a string formatted as address to an PlcBitAddress object.

C#

```
public static implicit operator PlcBitAddress(string value)
```

Exceptions

FormatException

The `value` is not a valid PLC byte address.

Inequality(PlcBitAddress, PlcBitAddress)

Returns a value indicating whether two instances of PlcBitAddress are not equal.

C#

```
public static bool operator !=(PlcBitAddress left, PlcBitAddress right)
```

LessThan(PlcBitAddress, PlcBitAddress)

Determines whether the first specified PlcBitAddress object is less than the second specified PlcBitAddress object.

C#

```
public static bool operator <(PlcBitAddress left, PlcBitAddress right)
```

Exceptions

ArgumentNullException

The `left` is a null reference (Nothing in Visual Basic).

LessThanOrEqual(PlcBitAddress, PlcBitAddress)

Determines whether the first specified PlcBitAddress object is less than or equal to the second PlcBitAddress object.

C#

```
public static bool operator <=(PlcBitAddress left, PlcBitAddress right)
```

Exceptions

ArgumentNullException

The `left` is a null reference (Nothing in Visual Basic).

Table of Contents

Constructors 1
PlcBitAddress(PlcOperand, Int32, Int32) 1
PlcBitAddress(PlcOperandType, Int32, Int32) 1
PlcBitAddress(PlcOperandType, Int32, Int32, Int32) 2
PlcBitAddress(SerializationInfo, StreamingContext) 3
Methods 3
CompareTo(Object) 3
CompareTo(PlcBitAddress) 4
Equals(Object) 4
Equals(PlcBitAddress) 5
GetHashCode() 5
Operators 5
Equality(PlcBitAddress, PlcBitAddress) 5
GreaterThan(PlcBitAddress, PlcBitAddress) 5
GreaterThanOrEqual(PlcBitAddress, PlcBitAddress) 6
Implicit(String to PlcBitAddress) 6
Inequality(PlcBitAddress, PlcBitAddress) 6
LessThan(PlcBitAddress, PlcBitAddress) 6
LessThanOrEqual(PlcBitAddress, PlcBitAddress) 7