

PlcByteAddress Members

Namespace: IPS7Lnk.Advanced

Assemblies: IPS7LnkNet.Advanced.dll, IPS7LnkNet.Advanced.dll

The [PlcByteAddress](#) type exposes the following members.

Constructors

PlcByteAddress(PlcOperand, Int32)

Initializes a new instance of the [PlcByteAddress](#) class using the specified **operand** and **byteNumber**.

C#

```
public PlcByteAddress(PlcOperand operand, int byteNumber)
```

Parameters

operand [PlcOperand](#)

The [PlcOperand](#) to that the new [PlcByteAddress](#) refers.

byteNumber [Int32](#)

The number of the byte to that the new [PlcByteAddress](#) refers.

Exceptions

[ArgumentException](#)

The type of **operand** specified is not supported.

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The **byteNumber** is out of the bounds defined by [MinByteNumber](#) or [MaxByteNumber](#).

PlcByteAddress(PlcOperandType, Int32)

Initializes a new instance of the [PlcByteAddress](#) class using the specified **operandType** and **byteNumber**.

C#

```
public PlcByteAddress(PlcOperandType operandType, int byteNumber)
```

Parameters

operandType [PlcOperandType](#)

The `PlcOperandType` to that the new `PlcByteAddress` refers.

`byteNumber` `Int32`

The number of the byte to that the new `PlcByteAddress` refers.

Exceptions

`ArgumentException`

The `operandType` specified requires an operand number.

`ArgumentOutOfRangeException`

The `byteNumber` is out of the bounds defined by `MinByteNumber` or `MaxByteNumber`.

PlcByteAddress(PlcOperandType, Int32, Int32)

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

C#

```
public PlcByteAddress(PlcOperandType operandType, int operandNumber, int byteNumber)
```

Parameters

`operandType` `PlcOperandType`

The `PlcOperandType` to that the new `PlcByteAddress` refers.

`operandNumber` `Int32`

The number of the operand to that the new `PlcByteAddress` refers.

`byteNumber` `Int32`

The number of the byte to that the new `PlcByteAddress` refers.

Exceptions

`ArgumentException`

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

`ArgumentOutOfRangeException`

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

PlcByteAddress(SerializationInfo, StreamingContext)

Initializes a new instance of the `PlcByteAddress` class with serialized data.

C#

```
protected PlcByteAddress(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

Bit(Int32)

Creates a new [PlcBitAddress](#) from this [PlcByteAddress](#) using the **bitNumber** specified.

C#

```
public PlcBitAddress Bit(int bitNumber)
```

Parameters

bitNumber [Int32](#)

The number of the bit to that the new [PlcBitAddress](#) refers.

Returns

[PlcBitAddress](#)

A new [PlcBitAddress](#) which refers to the **bitNumber** specified.

Exceptions

[ArgumentException](#)

The type of [Operand](#) does not support a bit number.

ArgumentOutOfRangeException

The `bitNumber` is out of the bounds defined by `MinBitNumber` or `MaxBitNumber`.

CompareTo(Object)

Compares the current `PlcByteAddress` with the `other`.

C#

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

Parameters

`other` Object

The `PlcByteAddress` to compare with this `PlcByteAddress`.

Returns

`Int32`

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

CompareTo(PlcByteAddress)

Compares the current `PlcByteAddress` with another `PlcByteAddress`.

C#

```
public int CompareTo(PlcByteAddress other)
```

Parameters

`other` `PlcByteAddress`

The `PlcByteAddress` to compare with this `PlcByteAddress`.

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 `PlcByteAddress`.

C#

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

Parameters

other Object

The [PlcByteAddress](#) to compare to the current [PlcByteAddress](#).

Returns

Boolean

The value true if the specified [PlcByteAddress](#) is equal to the current [PlcByteAddress](#); otherwise the value false.

Equals(PlcByteAddress)

Determines whether the specified **other** is equal to this [PlcByteAddress](#).

C#

```
public bool Equals(PlcByteAddress other)
```

Parameters

other [PlcByteAddress](#)

The [PlcByteAddress](#) to compare to the current [PlcByteAddress](#).

Returns

Boolean

The value true if the specified [PlcByteAddress](#) is equal to the current [PlcByteAddress](#); otherwise the value false.

GetHashCode()

Retrieves a hash code for this [PlcByteAddress](#).

C#

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

Returns

[Int32](#)

An [Int32](#) that contains the hash code for the [PlcByteAddress](#).

Operators

Equality(PlcByteAddress, PlcByteAddress)

Returns a value indicating whether two instance of [PlcByteAddress](#) are equal.

C#

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

GreaterThan(PlcByteAddress, PlcByteAddress)

Determines whether the first specified [PlcByteAddress](#) object is greater than the second specified [PlcByteAddress](#) object.

C#

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

GreaterThanOrEqual(PlcByteAddress, PlcByteAddress)

Determines whether the first specified [PlcByteAddress](#) object is greater than or equal to the second specified [PlcByteAddress](#) object.

C#

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

Implicit(String to PlcByteAddress)

Converts a string formatted as address to an [PlcByteAddress](#) object.

C#

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

Exceptions

[FormatException](#)

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

Inequality(PlcByteAddress, PlcByteAddress)

Returns a value indicating whether two instances of [PlcByteAddress](#) are not equal.

C#

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

LessThan(PlcByteAddress, PlcByteAddress)

Determines whether the first specified [PlcByteAddress](#) object is less than the second specified [PlcByteAddress](#) object.

C#

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

Exceptions

[ArgumentNullException](#)

The [left](#) is a null reference (Nothing in Visual Basic).

LessThanOrEqual(PlcByteAddress, PlcByteAddress)

Determines whether the first specified [PlcByteAddress](#) object is less than or equal to the second [PlcByteAddress](#) object.

C#

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

Exceptions

[ArgumentNullException](#)

The [left](#) is a null reference (Nothing in Visual Basic).

Table of Contents

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