

IP-S7-Link .Net Advanced MATLAB Code Snippets



Device Provider

Use one of the different PLC device providers by instantiating an instance of the appropriate PlcDevice class derivatives.

```
>> device = SiemensDevice(IPDeviceEndPoint('192.168.0.80'));
```

Device Configuration

After an instance of (for e.g.) the SiemensDevice class has been created just use the properties provided by the class to setup the appropriate device metadata.

```
>> device = SiemensDevice(IPDeviceEndPoint('192.168.0.80'));  
>> device.Type = SiemensDeviceType.S71200;
```

In the most scenarios its already enough to create a device using the appropriate device type in the constructor of the device, because the default channel does mostly match the common needs.

```
>> device = SiemensDevice(IPDeviceEndPoint('192.168.0.80'), SiemensDeviceType.S71200);
```

Device End Point

The toolbox does provide the ability to define different types of end points through the PlcDeviceEndPoint class. The most common end point implementation to use is the IPDeviceEndPoint class. Using this class it is possible to specify the IP address and optionally the rack and slot number of the PLC.

```
>> endPoint = IPDeviceEndPoint('192.168.0.80');  
  
% Alternatively also specify the rack.  
>> endPointWithRack = IPDeviceEndPoint('192.168.0.80', );  
  
% Alternatively also specify the rack and slot.  
>> endPointWithRackSlot = IPDeviceEndPoint('192.168.0.80', , 2);
```

Device Connection

After creating an instance of one of the PlcDevice class derivatives just create a new connection associated with the device represented. Creating an instance using this factory method will then return the device provider dependent implementation of the PlcDeviceConnection class.

```
>> device = SiemensDevice(IPDeviceEndPoint('192.168.0.80'));
>> connection = device.CreateConnection();
>> connection.Open();
```

Read Values

To read a single value use one of the Read methods of the PlcDeviceConnection class.

```
>> value = connection.ReadInt32('DB1.DBD 1')
```

To read an array of values use the additional read overloads to specify the number of items to read as an array as follows:

```
>> values = connection.ReadInt32('DB1.DBD 1', 3);
```

Read Values using PlcValue

To read a single value using an derivate of the PlcValue class ReadValues overloads of the PlcDeviceConnection class as follows.

```
>> value = PlcInt32('DB1.DBD 1');
>> valueData = connection.ReadValues({ value });
```

To read an array of values using an derivate of the PlcArray class use one of the ReadValues overloads as follows:

```
>> values = PlcInt32Array('DB1.DBD 1', 3);
>> valuesData = connection.ReadValues({ values });
```

To read multiple values at once using derivatives of the PlcValue class use one of the ReadValues overloads of the PlcDeviceConnection class as follows.

```
>> value1 = PlcInt32('DB1.DBD 1');
>> value2 = PlcInt32('DB2.DBD 1');
>> value3 = PlcInt32('DB3.DBD 1');

>> values = connection.ReadValues({ value1, value2, value3 });
```

To read multiple arrays of values at once using derivatives of the PlcArray class use one of the ReadValues overloads of the PlcDeviceConnection class as follows.

```
>> values1 = PlcInt32Array('DB1.DBD 1', 3);
>> values2 = PlcInt32Array('DB2.DBD 1', 4);
>> values3 = PlcInt32Array('DB3.DBD 1', 5);

>> values = connection.ReadValues({ values1, values2, values3 });
```

To read multiple derivatives of the PlcValue class or of the PlcArray class use one of the ReadValues overloads of the PlcDeviceConnection class as follows.

```
>> value1 = PlcInt32('DB1.DBD 1');
>> values2 = PlcInt32Array('DB2.DBD 1', 4);
>> value3 = PlcBoolean('DB3.DBX 1.0');
>> values4 = PlcBooleanArray('DB4.DBX 1.0', 5);
>> value5 = PlcString('DB5.DBB 1', 32);

>> values = connection.ReadValues({ value1, values2, value3, values4, value5 });
```

Write Values

To write a single value use one of the Write methods of the PlcDeviceConnection class.

```
>> connection.WriteInt32('DB1.DBD 1', 123);
```

To write an array of values use the additional write overloads to specify an array of items to write as follows:

```
>> connection.WriteInt32('DB1.DBD 1', [ 123 456 789 ]);
```

Write Values using PlcValue

To write a single value using an derivate of the PlcValue class use one of the WriteValues overloads of the PlcDeviceConnection class as follows.

```
>> value = PlcInt32('DB1.DBD 1', 123);
>> connection.WriteValues({ value });
```

To write an array of values using an derivate of the PlcArray class use one of the WriteValues overloads as follows:

```
>> values = PlcInt32Array('DB1.DBD 1', [ 123 456 789 ]);
>> connection.WriteValues({ values });
```

To write multiple values at once using derivatives of the PlcValue class use one of the WriteValues overloads of the PlcDeviceConnection class as follows.

```
>> value1 = PlcInt32('DB1.DBD 1', 123);
>> value2 = PlcInt32('DB2.DBD 1', 456);
>> value3 = PlcInt32('DB3.DBD 1', 789);

>> connection.WriteValues({ value1, value2, value3 });
```

To write multiple arrays of values at once using derivatives of the PlcArray class use one of the WriteValues overloads of the PlcDeviceConnection class as follows.

```
>> values1 = PlcInt32Array('DB1.DBD 1', [ 1 2 3 ]);
>> values2 = PlcInt32Array('DB2.DBD 1', [ 4 5 6 ]);
>> values3 = PlcInt32Array('DB3.DBD 1', [ 7 8 9 ]);

>> connection.WriteValues({ values1, values2, values3 });
```

To write multiple derivatives of the PlcValue class or of the PlcArray class use one of the WriteValues

overloads of the PlcDeviceConnection class as follows.

```
>> value1 = PlcInt32('DB1.DBD 1', 123);  
>> values2 = PlcInt32Array('DB2.DBD 1', [ 4 5 6 ]);  
>> value3 = PlcBoolean('DB3.DBX 1.0', true);  
>> values4 = PlcBooleanArray('DB4.DBX 1.0', [ true false true false true ]);  
>> value5 = PlcString('DB5.DBB 1', 'This is my string value.');
```

```
>> connection.WriteValues({ value1, values2, value3, values4, value5 });
```

Table of Contents

Device Provider	1
Device Configuration	1
Device End Point	1
Device Connection	1
Read Values	2
Read Values using PlcValue	2
Write Values	3
Write Values using PlcValue	3

