

IP-S7-Link .Net Advanced MATLAB Code Snippets



Device Provider

Verwenden Sie einen der verschiedenen SPS-Geräteanbieter, indem Sie eine Instanz der entsprechenden Ableitung von der PlcDevice Klasse erstellen.

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

Device Konfiguration

Nachdem eine Instanz von (z. B.) der SiemensDevice-Klasse erstellt wurde, verwenden Sie einfach die bereitgestellten Properties, um die entsprechenden Geräte-Metadaten einzurichten.

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

In den meisten Szenarien ist es schon genug, im Konstruktor den entsprechenden Gerätetypen zu übergeben, da der Default-Channel meistens den gemeinsamen Anforderungen entspricht.

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

Device End Point

Die Toolbox bietet die Möglichkeit, verschiedene Typen von End Points über die PlcDeviceEndPoint-Klasse zu definieren. Die am häufigsten verwendete End Point-Implementierung ist die IPDeviceEndPoint-Klasse. Mit dieser Klasse können Sie die IP-Adresse und optional die Rack- und Slotnummer der SPS angeben.

```
>> 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

Nach dem Erstellen der Instanz, generieren Sie einfach eine neue Verbindung, die mit dem dargestellten Gerät verknüpft ist. Wenn Sie eine Instanz mit dieser Factory-Methode erstellen, wird die vom Provider abhängige Implementierung der PlcDeviceConnection-Klasse zurückgegeben.

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

Werte lesen

Um einen einzelnen Wert zu lesen, verwenden Sie eine der Read-Methoden der PlcDeviceConnection-Klasse.

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

Um ein Array von Werten zu lesen, verwenden Sie die zusätzlichen Read-Überladungen, um die Anzahl der zu lesenden Elemente als Array wie folgt anzugeben:

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

Werte mit PlcValue schreiben

Einzelnen Wert lesen:

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

Ein Array von Werten lesen:

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

Mehrere Werte lesen:

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

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

Mehrere Arrays lesen:

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

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

Unterschiedliche Werttypen lesen:

```
>> 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 });
```

Werte schreiben

Einzelnen Wert schreiben:

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

Array schreiben:

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

Werte mit PlcValue schreiben

Einen Wert schreiben:

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

Array schreiben:

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

Mehrere Werte schreiben:

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

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

Mehrere Arrays schreiben:

```
>> 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 });
```

Unterschiedliche Werte schreiben:

```
>> 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 });
```

Inhaltsverzeichnis

Device Provider	1
Device Konfiguration	1
Device End Point	1
Device Connection	1
Werte lesen	2
Werte mit PlcValue schreiben	2
Werte schreiben	3
Werte mit PlcValue schreiben	3

