

Zugriff auf den Miniserver

LoxBerry XL stellt beim Starten des Scripts automatisch die Verbindung zu deinem bzw. deinen Miniservern her. Es werden dafür die Angaben und Einstellungen im Miniserver-Widget von LoxBerry genutzt.

In deinem Script hast du damit automatisch Zugriff mit der Variable **\$ms1**, bei mehreren Miniservern z.B. **\$ms1**, **\$ms2**, **\$ms3** und so weiter. Beim Starten des Scripts wird angezeigt, welche Miniserver verfügbar sind:

```
LoxBerry XL Version 2.0.2.6
Miniserver 1 (MSOG) accessible by $ms1
Miniserver 2 (MSUG) accessible by $ms2
Miniserver 3 (TestMS) accessible by $ms3
Miniserver 4 (Test-MSv2) accessible by $ms4
```

Mit dieser Variable hast du dann **direkten Zugriff** auf die Logik-Bausteine und Eingänge/Ausgänge deines Miniservers:

Beispiel

The image shows a configuration window for an 'EIB-Taster' object named 'LED_Spot'. The 'Bezeichnung' (Label) is 'LED_Spot' and the 'Beschreibung' (Description) is 'LED Spot'. Below this is a terminal window showing the command `echo "Status LED-Spot: " . $ms3->LED_Spot . "\n";` and its output: `Status LED-Spot: 1`. A second terminal window shows the list of available miniservers, including `$ms3`.

Bezeichnung der Ein-/Ausgänge des Miniservers

Übergeben wird im Beispiel bei **\$ms3->LED_Spot** **exakt** die Bezeichnung des Bausteins, oder Virtuellen Eingangs, Virtuellen Texteingangs oder Virtuellen Ausgangs. Bezeichnung und Beschreibung!? → "Bezeichnung" und "Beschreibung" von Objekten

Zu beachten gilt dabei Folgendes:

- Bezeichnungen in der Loxone Config müssen **eindeutig** sein (wenn alle Bausteine "Lichtschalter" heißen, nimmt der Miniserver einfach irgendeinen davon)
- Im Gateway-Konzentrator-Betrieb muss der Befehl an den tatsächlichen Miniserver gesendet werden, nicht an den Konzentrador
- Die Bezeichnungen sind case-sensitive (heißt der Baustein "Außentemperatur", muss auch "Außentemperatur" verwendet werden, nicht "außentemperatur").
- In der Standard-Syntax sind **erlaubt** : Umlaute und z.B. das scharfe ß
- In der Standard-Syntax sind **nicht erlaubt** : Alle von PHP als Steuerzeichen gewertete Zeichen,

insbesondere Leerzeichen, Bindestrich (-), Beistrich (,), Dollar-Zeichen (\$), Größer/Kleiner-Zeichen (</>)

- Dafür gibt es Spezialsyntaxen, um diese Bausteine trotzdem ansprechen zu können.
- Optimal ist es, wenn die Bezeichnung komplett ohne Leerzeichen und Sonderzeichen auskommt, und für die Visualisierung die Beschreibung definiert wird. "**Bezeichnung**" und "**Beschreibung**" von Objekten

Folgende Syntax **funktioniert**:

- `$ms1->LED_Spot` → Underscores erlaubt
- `$ms1->Süden_Außentemperatur` → Umlaute erlaubt
- `$ms1->{'LED-Spot'}` → Spezialsyntax, um Leerzeichen, Bindestriche und Sonderzeichen angeben zu können
- `$ms1->get('LED Spot')` → Spezialsyntax, um Leerzeichen, Bindestriche und Sonderzeichen angeben zu können

Folgende Syntax **funktioniert nicht**:

- `$ms1->LED Spot` → Leerzeichen nicht erlaubt!
- `$ms1->LED-Spot` → Bindestrich nicht erlaubt!

Abrufen und Senden von Daten vom/zum Miniserver

Die Syntaxen zum Abrufen und Senden von Daten sind im Beispiel gezeigt.

Beim **Senden** von Daten können sowohl dezimale Werte (auch mit Nachkommastellen), Texte, als auch Digitalwerte gesendet werden.

Für digitale Bausteine stehen die Befehle der [Loxone Webservices](#) zur Verfügung: **Ein**, **On**, **Aus**, **Off**, **Impuls**, **Pulse**, sowie für Bausteine mit zwei Ausgängen: **ImpulsPlus**, **ImpulsMinus**, **PulseUp**, **PulseDown**, **ImpulsAuf**, **ImpulsAb**, **PulseOpen**, **PulseClose**, **PlusEin**, **PlusAus**, **UpOn**, **UpOff**, **AufEin**, **AufAus**, **OpenOn**, **OpenOff**, **MinusEin**, **MinusAus**, **DownOn**, **DownOff**, **AbEin**, **AbAus**, **CloseOn**, **CloseOff**

Syntax- und Code-Beispiele

```
// Daten abrufen
$licht_ist_an = $ms1->LED_Spot;
$licht_ist_an = $ms1->LED_Spot();
$licht_ist_an = $ms1->{'LED-Spot'};
$licht_ist_an = $ms1->get('LED-Spot');

// Daten senden
$ms1->LED_Spot("On");
$ms1->LED_Spot = "On";
$ms1->{'LED-Spot'} = "On";
$ms1->set('LED-Spot', "On");
```

```
$ms1->Alarmtext = "Der Alarm wurde ausgelöst!";
$ms1->Soll_Temperatur = 23.5;
```

Einheit filtern

Anzeige	
Einheit	<v.1>°C
Eingabetype	Schieber
Schrittweite	1
<input type="checkbox"/> nur Statusanzeige	

Das Abrufen von Daten liefert eine gegebenenfalls in den Eigenschaften des Bausteins hinterlegte Einheit mit.

In diesem Beispiel würde der Abruf `$ms1->Außentemperatur` zurückliefern: **7.5°C**

Um die Einheit zu filtern, liefert **LoxBerry XL** die Funktion `clean()` mit:

```
echo $ms1->Außentemperatur;
// Ausgabe: 7.5°C

echo clean( $ms1->Außentemperatur );
// Ausgabe: 7.5
```

Vorsicht bei Schleifen und if-Abfragen

Schlechtes Beispiel :

```
if ( $ms1->Außentemperatur < -20 ) {
    echo "Heute ist es sehr, sehr kalt!";
} elseif( $ms1->Außentemperatur < 0 ) {
    echo "Es gefriert draußen.";
} elseif( $ms1->Außentemperatur < 20 ) {
    echo "Temperatur ist ganz ok";
} else {
    echo "Heiß heute!";
}
```

Dieser Code führt drei Abfragen gegen den Miniserver aus. (Randnotiz: Eventuell implementieren wir noch einen Kurzzeit-Cache, um das zu verbessern.)

Gleiche Funktion, aber Gutes Beispiel (1) :

```
$temp = $ms1->Außentemperatur;
if ( $temp < -20 ) {
    echo "Heute ist es sehr, sehr kalt!";
} elseif( $temp < 0 ) {
    echo "Es gefriert draußen.";
} elseif( $temp < 20 ) {
```

```
    echo "Temperatur ist ganz ok";  
} else {  
    echo "Heiß heute!";  
}
```

Das braucht nur noch eine Abfrage.

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

https://wiki.loxberry.de/konfiguration/loxberry_xl_extended_logic/zugriff_auf_den_miniserver

Last update: **2022/09/10 12:18**