



LCU1 · Linux Control Unit

Logik-Switch (icserver)

Inhalt

1	Einführung.....	3
2	Konfigurationsdateien.....	3
2.1	Schleifenmechanismus (<loop>-Element).....	3
2.1.1	Element <loop>.....	3
2.1.1.1	Attribute des <loop>-Elements.....	3
2.1.1.2	Beispiel.....	4
2.2	Systemkonfiguration.....	4
2.2.1	Root-Element <icserver_system>.....	4
2.2.1.1	Untergeordnete Elemente des <icserver_system>-Elements.....	4
2.2.2	Element <devices>.....	4
2.2.2.1	Untergeordnete Elemente des <devices>-Elements.....	4
2.2.3	Element <device>.....	5
2.2.3.1	Attribute des <device>-Elements.....	5
2.2.3.2	Untergeordnete Elemente des <device>-Elements.....	5
2.2.4	Element <channel>.....	5
2.2.4.1	Attribute des <channel>-Elements.....	5
2.2.4.2	Untergeordnete Elemente des <channel>-Elements.....	6
2.2.5	Element <value> (Unterelement von <channel>).....	6
2.2.5.1	Attribute des <value>-Elements.....	6
2.2.6	Element <values>.....	6
2.2.6.1	Untergeordnete Elemente des <values>-Elements.....	6
2.2.7	Element <value> (Unterelement von <values>).....	6
2.2.7.1	Attribute des <value>-Elements.....	7
2.2.8	Element <interfaces>.....	7
2.2.8.1	Untergeordnete Elemente des <interfaces>-Elements.....	8
2.2.9	Element <interface>.....	8
2.2.9.1	Attribute des <interface>-Elements.....	8
2.3	Benutzerkonfiguration.....	8
2.3.1	Root-Element <icserver_user>.....	8
2.3.1.1	Untergeordnete Elemente des <icserver_user>-Elements.....	8
2.3.2	Element <devices>.....	8
2.3.2.1	Untergeordnete Elemente des <devices>-Elements.....	9
2.3.3	Element <device>.....	9
2.3.3.1	Attribute des <device>-Elements.....	9
2.4	Persistierung.....	9

1 Einführung

Dieses Dokument beschreibt den Logik-Switch der LCU1.

Der Logik-Switch ist die Softwareinstanz, in der die Logiksignale "leben". Jedem Logiksignal können Kanalwerte im HomeMatic XML-RPC-Protokoll zugeordnet werden. Eine Änderung des Logiksignals wirkt sich dann auch auf einen als Ziel zugeordneten Kanalwert aus. Umgekehrt wirkt sich eine Änderung eines Quell-Kanalwertes unmittelbar auf das Logiksignal aus.

Beliebig viele HomeMatic-Interfaceprozesse können per XML-RPC an den Logik-Switch angebunden werden.

Neben dem HomeMatic XML-RPC-Protokoll unterstützt der Logik-Switch das von immediateC definierte Netzwerkprotokoll für Logiksignale. Über dieses immediateC-Protokoll verbinden sich die UI-Engine sowie ein oder mehrere Logikprogramme mit dem Logik-Switch.

Der Logikswitch kann optional ausgewählte Logiksignale persistieren. Diese stehen dann nach einem Neustart der LCU mit dem alten Wert zur Verfügung.

Der Logikswitch ist in der Firmware der LCU1 enthalten. Die ausführbare Datei liegt unter `/bin/icserver`. Der Logikswitch wird beim Start der LCU automatisch gestartet.

2 Konfigurationsdateien

2.1 Schleifenmechanismus (`<loop>`-Element)

In den nachfolgend beschriebenen XML-Dateien ist es an einigen Stellen möglich, mittels eines Schleifenkonstruktes Schreibarbeit zu sparen.

Dabei kommt das `<loop>`-Element zum Einsatz.

2.1.1 Element `<loop>`

Das `<loop>`-Element arbeitet wie eine Programmschleife, die für die mit dem Attribut `count` festgelegte Anzahl von Zyklen durchlaufen wird. Die Laufvariable ist immer „n“ und wird beginnend bei dem mit dem Attribut `start` festgelegten Wert mit jedem Durchlauf inkrementiert. Innerhalb der Schleife können in Attributen mit `[]` markierte Ausdrücke verwendet werden, die auf „n“ basieren. Die gleiche Syntax für Schleifen wird auch von immediateC-Arrays verwendet (siehe immediateC-Dokumentation).

Als untergeordnete Elemente des `<loop>`-Elementes sind immer genau die Elemente zulässig, die als untergeordnete Elemente des umgebenden Elementes zulässig sind.

2.1.1.1 Attribute des `<loop>`-Elements

- `start [integer, Pflicht]`
Dieses Attribut ist verpflichtend. Es gibt den Startindex für den Schleifenzähler an.
- `count [integer, Pflicht]`
Dieses Attribut ist verpflichtend. Es gibt die Anzahl der Schleifendurchläufe an.

2.1.1.2 Beispiel

```
<values>
  <!-- On board inputs -->
  <loop start="0" count="6">
    <value name="IX1.[n]" alias="binInOnboardInput_[n]" />
  </loop>

  <!-- On board buzzer -->
  <value name="QX1.5" alias="binOutBuzzer" />

  <!-- On board Relay -->
  <loop start="0" count="4">
    <value name="QX1.[n]" alias="binOutOnboardRelay_[n]" />
  </loop>
</values>
```

2.2 Systemkonfiguration

Die Systemkonfiguration befindet sich in der Datei

/etc/icserver/icserver_system.conf im schreibgeschützten Flashbereich.

Diese Datei enthält die grundlegende vom Endanwender nicht veränderbare Konfiguration. Dies umfasst:

- die Definition der fest in der Applikation vorhandenen Logiksignale
- die Definition der fest vorhandenen XML-RPC Interfaceprozesse, z.B. des Plattformprozesses (pfmd)
- die Definition der über XML-RPC angebundenen Geräte und Kanäle mit Information zur Zuordnung von Logiksignalen zu Kanalwerten
- die zum Ansprechen der XML-RPC-Geräte und Kanäle notwendige Information zu Interfaceprozessen und Geräteadressen. Diese Information ist in dieser Datei nur für die Geräte mit einer festen Adresse und einem festen Interfaceprozess vorhanden. Im speziellen ist dies das Systemgerät mit den Ein- und Ausgängen der LCU.

2.2.1 Überblick anhand eines Beispiels

Es wird im Folgenden anhand eines Beispiels für die Systemkonfiguration ein Überblick gegeben.

2.2.1.1 Zuordnung der XML-RPC-Kanäle

Die folgende Textbox zeigt die Definition der Kanäle des vom pfmd bereitgestellten „SYSTEM“-Gerätes und die Zuordnung zu Logiksignalen.

```

<icserver_system>
  <devices>
    <device id="INOUT" type="SYSTEM" interface="SYSTEM"
      address="SYSTEM">
      <channel index="1" terminal="KL1.23">
        <value id="STATE" target="binInOnboardInput_0"/>
      </channel>
    [...]
      <channel index="5" terminal="Taster oben">
        <value id="STATE" target="binInOnboardInput_4"/>
      </channel>
    [...]
      <channel index="7" terminal="KL1.9 - KL1.11">
        <value id="STATE" source="binOutOnboardRelay_0"/>
      </channel>
    [...]
      <channel index="11" terminal="Piezo">
        <value id="STATE" source="binOutBuzzer"/>
      </channel>
      <channel index="12" terminal="ADC0">
        <value id="VALUE" target="intInOnboardADC_0"/>
      </channel>
    [...]
      <channel index="13" terminal="ADC1">
        <value id="VALUE" target="intInOnboardADC_1"/>
      </channel>
    </device>
  </devices>
[...]
```

Unter <devices> sind alle über XML-RPC angebotenen Geräte aufgelistet. <device> definiert das „SYSTEM“-Gerät des pfm. Es wird auf das Interface „SYSTEM“ verwiesen, das weiter unten in der Datei definiert ist.

Unterhalb von <device> werden die für die Logiksignale relevanten Kanäle jeweils als ein Element <channel> aufgelistet.

Das Attribut index des Elementes <channel> gibt die XML-RPC-Kanalnummer an, das Attribut id des Elementes <value> gibt die Id des XML-RPC-Kanalwertes an. Die Attribute source für Ausgänge bzw. target für Eingänge stellen die Verbindung zu den weiter unten definierten Logiksignalen her.

Es werden also die Digitaleingänge der LCU auf die Logiksignale binInOnboardInput_0 bis binInOnboardInput_3 gelegt. Die beiden Taster werden auf binInOnboardInput_4 und binInOnboardInput_3 gelegt. Die Relais werden zu binOutOnboardRelay_0 bis binOutOnboardRelay_3. Der Signalgeber wird binOutBuzzer und die ADC-Eingänge werden intInOnboardADC_0 und intInOnboardADC_1.

2.2.1.2 Definition der Logiksignale

Als Nächstes werden die oben bereits verwendeten Logiksignale definiert:

Unterhalb des Elementes <values> werden in Form von <value>-Elementen alle Logiksignale aufgelistet. Dabei kann mit dem <loop>-Element Schreibarbeit gespart werden. Das <loop>-Element arbeitet wie eine Programmschleife, die count Zyklen durchlaufen wird. Die Laufvariable ist immer „n“ und wird beginnend bei start mit jedem Durchlauf incrementiert. Innerhalb der Schleife können in Attributen mit [] markierte Ausdrücke verwendet werden, die auf „n“ basieren.

```
[...]
  <values>
    <!-- On board inputs -->
    <loop start="0" count="6">
      <value name="IX1.[n]" alias="binInOnboardInput_[n]"/>
    </loop>

    <!-- On board Relay -->
    <loop start="0" count="4">
      <value name="QX1.[n]" alias="binOutOnboardRelay_[n]"/>
    </loop>

    <!-- On board buzzer -->
    <value name="QX1.5" alias="binOutBuzzer" iccignore="true"/>

    <!-- On board ADCs -->
    <loop start="0" count="2">
      <value name="IL[1+n]" alias="intInOnboardADC_[n]"/>
    </loop>

  </values>
[...]
```

Die gleiche Syntax für Schleifen wird auch von immediateC-Arrays verwendet (siehe immediateC-Dokumentation).

Für jedes Logiksignal wird ein Name in IEC-1131-Notation und ein sprechender Alias angegeben. Oben in dem Abschnitt mit der XML-RPC-Anbindung werden die Logiksignale über die Aliasnamen referenziert.

So werden z.B. den Logiksignalen IX1.0 bis IX1.5 die Aliasnamen binInOnboardInput_0 bis binInOnboardInput_5 zugeordnet, welche über die obige XML-RPC-Zuordnung wiederum den 4 Digitaleingängen und den beiden Tastern entsprechen.

Das Attribut `iccignore="true"` am Signal binOutBuzzer sorgt dafür, dass `xml2iha.tcl` (siehe oben) für dieses Signal keine Definition für die Logiksteuerung in der Datei `external_signals.iha` erzeugt. Der Buzzer wird nämlich von dem unten beschriebenen Logikprogramm nicht verwendet und auf diesem Wege wird eine Fehlermeldung der immediateC-Compilers für einen undefinierten Ausgang umgangen.

2.2.1.3 Definition der XML-RPC-Interfaceprozesse

Jetzt muss `icserver` nur noch wissen, wie der XML-RPC-Interfaceprozess für das Gerät „SYSTEM“ angesprochen werden kann:

```
[...]
  <interfaces>
    <interface id="SYSTEM" url="bin://127.0.0.1:2002"/>
  </interfaces>
</icserver_system>
```

Hier wird ein Interface mit dem Namen „SYSTEM“ definiert, das auf dem lokalen Rechner (IP-Adresse 127.0.0.1) auf dem Port 2002 angesprochen wird.

2.2.2 Formale Beschreibung des XML-Formates

2.2.2.1 Root-Element `<icserver_system>`

Das Rootelement der Systemkonfiguration ist immer `<icserver_system>`.

2.2.2.1.1 Untergeordnete Elemente des `<icserver_system>`-Elements

- `devices`
Dieses Element enthält eine Liste der XML-RPC-Geräte, deren Kanäle zu Logiksignalen zugeordnet sind. Siehe 2.2.3.
- `values`
Dieses Element enthält eine Liste der Logiksignale. Siehe 2.2.5.2.
- `interfaces`
Dieses Element enthält eine Liste der an den Logik-Switch angebotenen XML-RPC-Interfaceprozesse. Siehe 2.2.5.4.

2.2.3 Element `<devices>`

Dieses Element enthält eine Liste der über XML-RPC anzusprechenden Geräte in Form von `<device>`-Elementen. Dieses Element unterstützt den Schleifenmechanismus (siehe 2.1)

2.2.3.1 Untergeordnete Elemente des `<devices>`-Elements

- `<loop>`
Steuerelement für den Schleifenmechanismus, siehe 2.1.
- `<device>`
Dieses Element beschreibt die Kanalwerte für ein über XML-RPC angebotenes Gerät. Siehe 2.2.3.2.

2.2.3.2 Element `<device>`

Dieses Element beschreibt die Zuordnungsinformationen für ein über XML-RPC angebotenes Gerät.

2.2.3.2.1 Attribute des `<device>`-Elements

- `id` [String, Pflicht]
Dieses Attribut ist verpflichtend. Es weist dem Gerät eine Id zu, über die es zur Zuordnung von logischen Geräten zu physikalischen Geräten referenziert werden kann. Innerhalb einer Schleife (`<loop>`) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.
- `type` [String, Pflicht]
Dieses Attribut ist verpflichtend. Es gibt den HomeMatic-Gerätetyp an, z.B. "HMW-IO-12-Sw14-DR".
- `display_name` [String, optional]
Dieses Attribut gibt den menschenlesbaren Namen des Gerätes an. Dieser wird in der Webseite zur Zuordnung von physikalischen Geräten zu logischen Geräten angezeigt. Innerhalb einer Schleife (`<loop>`) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.
- `interface` [String, optional]
Dieses Attribut verweist auf die Id des XML-RPC-Interfaceprozesses, über den das physikalische Gerät angesprochen wird. Dieses Attribut ist Teil der Zuordnungsinformation und sollte hier nur für feste Geräte mit unveränderlichem Interfaceprozess angegeben werden. Für andere Geräte siehe 2.3.

- `address [String, Pflicht wenn interface angegeben ist]`
Dieses Attribut gibt die Adresse (Seriennummer) des Gerätes an. Dieses Attribut ist Teil der Zuordnungsinformation und sollte hier nur für feste Geräte mit unveränderlicher Adresse angegeben werden. Für andere Geräte siehe 2.3.

2.2.3.2.2 Untergeordnete Elemente des `<device>`-Elements

- `channel`
Dieses Element definiert einen Kanal eines logischen über XML-RPC angebundenen Gerätes. Siehe 2.2.3.3.

2.2.3.3 Element `<channel>`

Dieses Element definiert einen Kanal eines logischen über XML-RPC angebundenen Gerätes.

2.2.3.3.1 Attribute des `<channel>`-Elements

- `index [integer, Pflicht]`
Dieses Attribut ist verpflichtend. Es gibt die XML-RPC-Kanalnummer an. Innerhalb einer Schleife (`<loop>`) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.
- `terminal [String, optional]`
Dieses Attribut gibt einen lesbaren Namen für den physikalischen Anschluss des Gerätes an, auf den sich der Kanal bezieht. Dieser Name wird auf der Konfigurations-Webseite mit dem Verdrahtungsplan ausgegeben. Innerhalb einer Schleife (`<loop>`) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.

2.2.3.3.2 Untergeordnete Elemente des `<channel>`-Elements

- `<value>`
Enthält die Zuordnungsinformationen zu einem XML-RPC-Kanalwert.
- `<configuration>`
Enthält eine Auflistung der Konfigurationsparameter des MASTER-Parametersets des XML-RPC-Kanals. Der Logik-Switch stellt sicher, dass die hier angegebenen Konfigurationsparameter die angegebenen Werte haben.

2.2.4 Element `<value>` (Unterelement von `<channel>`)

Dieses Element definiert die Zuordnungsinformationen für einen XML-RPC-Kanalwert.

2.2.4.1 Attribute des `<value>`-Elements

- `id [String, Pflicht]`
Dieses Attribut ist verpflichtend. Es gibt die XML-RPC-Id des Wertes an.
- `float_scale [float, optional, default 100.0]`
Dieses Attribut gibt den Skalierungsfaktor an für den Fall, dass der XML-RPC-Kanalwert vom Typ Float ist. Der Skalierungsfaktor wird verwendet, um den XML-RPC-Fließkommawert in einen immediateC-Integerwert umzurechnen. Der XML-RPC-Wert des Gerätes wird dabei mit diesem Faktor multipliziert, um den immediateC-Wert zu erhalten.
- `bool_invert [Boolean, optional, default false]`
Dieses Attribut gibt an, ob für den Fall, dass der XML-RPC-Kanalwert vom Typ Boolean ist, der Wert bei der Übertragung zwischen XML-RPC und immediateC invertiert werden soll.
- `source [String, Pflicht wenn "target" nicht angegeben ist]`

Dieses Attribut gibt den Namen oder Alias eines immediateC-Logiksignals an, mit dem dieser XML-RPC-Kanalwert verbunden wird. Eine Änderung des Logiksignals löst dabei eine Änderung des XML-RPC-Kanalwertes aus.

Innerhalb einer Schleife (<loop>) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.

- target [String, Pflicht wenn "source" nicht angegeben ist]

Dieses Attribut gibt den Namen oder Alias eines immediateC-Logiksignals an, mit dem dieser XML-RPC-Kanalwert verbunden wird. Eine Änderung des XML-RPC-Kanalwertes löst dabei eine Änderung des Logiksignals aus.

Innerhalb einer Schleife (<loop>) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.

2.2.4.2 Element <configuration>

Dieses Element enthält eine Auflistung der Konfigurationsparameter des MASTER-Parametersets des XML-RPC-Kanals. Der Logik-Switch stellt sicher, dass die hier angegebenen Konfigurationsparameter die angegebenen Werte haben.

2.2.4.2.1 Untergeordnete Elemente des <configuration>-Elements

- <parameter>

Enthält die Konfigurationsinformationen für einen XML-RPC-Kanalparameter.

2.2.5 Element <parameter>

Dieses Element enthält die Konfigurationsinformationen für einen XML-RPC-Kanalparameter.

2.2.5.1 Attribute des <parameter>-Elements

- name [String, Pflicht]
Dieses Attribut ist verpflichtend. Es gibt den XML-RPC-Parameternamen an.
- value [String, optional]
Dieses Attribut gibt den Wert an, auf den der Konfigurationsparameter gesetzt werden soll.
- reset_to_default [Boolean, optional]
Wenn dieses Attribut gesetzt ist, wird der Parameter auf den in der Gerätebeschreibung definierten Defaultwert gesetzt.

2.2.5.2 Element <values>

Dieses Element enthält eine Liste der immediateC-Werte in Form von <value>-Elementen.

Dieses Element unterstützt den Schleifenmechanismus (siehe 2.1)

2.2.5.2.1 Untergeordnete Elemente des <values>-Elements

- <loop>
Steuerelement für den Schleifenmechanismus, siehe 2.1.
- <value>
Dieses Element beschreibt ein immediateC-Logiksignal. Siehe 2.2.3.2.

2.2.5.3 Element <value> (Unterelement von <values>)

Dieses Element beschreibt ein immediateC-Logiksignal. Für jedes Logiksignal muss ein Name in IEC-1131-Notation angegeben werden. Zusätzlich kann ein sprechender Alias spezifiziert werden. Laut IEC-1131 bzw. EN-61131 beginnen Eingänge mit "I" und Ausgänge mit "Q". Der nächste Buchstabe gibt an, ob es sich um ein Bit ("X"), ein Byte ("B"), ein 16bit-Wort ("W") oder ein 32bit-Wort ("L") handelt. Danach folgt eine Nummer und bei einem Bit noch durch "." getrennt eine

Bitnummer.

IX1.2 ist somit ein Bit-Eingang, QB6 ist ein 8bit-Ausgang und IL11 ist ein 32bit-Eingang.

2.2.5.3.1 Attribute des **<value>-Elements**

- `name [String, Pflicht]`
Dieses Attribut ist verpflichtend. Es weist dem Logiksignal einen IEC-1131-Namen zu. Die Einhaltung der IEC-1131-Konvention ist wichtig, damit die Zusammenarbeit mit immediateC richtig funktioniert.
Innerhalb einer Schleife (`<loop>`) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.
- `alias [String, optional]`
Dieses Attribut weist dem Logiksignal einen zweiten Aliasnamen zu. Der Aliasname unterliegt keinen Restriktionen. Über diesen Alias kann das Logiksignal in der Programmierung der Oberfläche verwendet werden. Über das Script `iclogic/xml2iha.tcl` wird beim Builden des Logikprogramms die Datei `external_signals.iha` erzeugt, welche die Aliasnamen der Signale auf die IEC-1131-Namen der Signale abbildet. Durch diesen Trick kann im Logikprogramm ebenfalls mit den Aliasnamen programmiert werden.
Innerhalb einer Schleife (`<loop>`) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.
- `persistent [Boolean, optional]`
Dieses Attribut gibt an, ob der Wert des Logiksignals persistiert werden soll. Die Werte von persistierten Logiksignalen werden in der Datei `/etc/config/icserver_values` abgelegt und aktuell gehalten. Die Persistierung bietet sich für Logiksignale an, die Konfigurationsoptionen an der Benutzeroberfläche entsprechen. Es sollten keine sich häufig ändernden Werte persistiert werden, weil dies negative Auswirkungen auf die Performance haben kann.
- `icignore [Boolean, optional]`
Dieses Attribut wird vom Logik-Switch nicht ausgewertet. Das Skript `iclogic/xml2iha.tcl` ignoriert die Logiksignale, bei denen dieses Attribut `true` ist. Das kann nötig sein, um für im Logikprogramm nicht verwendete Ausgänge die Fehlermeldung des Compilers zu vermeiden.
- `display_name [String, optional]`
Dieses Attribut weist dem Logiksignal einen menschenlesbaren Namen für die Anzeige auf Konfigurations-Webseiten zu. Dieser Name wird z.B. auf der Webseite für den Verdrahtungsplan verwendet.
Innerhalb einer Schleife (`<loop>`) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.
- `default_value [integer, optional, default=0]`
Dieses Attribut weist dem Logiksignal einen Standardwert zu, der nach dem Start des Logik-Switches angenommen wird.

2.2.5.4 Element **<interfaces>**

Dieses Element enthält eine Liste der fest im System vorhandenen und vom Benutzer nicht veränderbaren XML-RPC-Interfaceprozesse.

2.2.5.4.1 Untergeordnete Elemente des **<interfaces>-Elements**

- `<interface>`
Dieses Element enthält die für die Kommunikation mit einem speziellen XML-RPC-Interfaceprozess nötigen Informationen. Siehe 2.2.5.5.

2.2.5.5 Element **<interface>**

Dieses Element enthält die für die Kommunikation mit einem speziellen XML-RPC-Interfaceprozess nötigen Informationen.

2.2.5.5.1 *Attribute des <interface>-Elements*

- `id [String, Pflicht]`
Dieses Attribut ist verpflichtend. Es weist dem Interfaceprozess eine Id zu. Über diese Id kann der Interfaceprozess aus der Gerätezuordnung (siehe 2.3.3.1) heraus referenziert werden.
- `url [String, Pflicht]`
Dieses Attribut ist verpflichtend. Es gibt die URL an, unter der der Interfaceprozess angesprochen werden kann.

2.3 Benutzerkonfiguration

Die Benutzerkonfiguration befindet sich in der Datei `/etc/config/icserver_user.conf` im schreibbaren Flashbereich.

Diese Datei enthält die vom Endanwender veränderbare Konfiguration. Dies umfasst:

- die Definition der vom Benutzer hinzugefügten XML-RPC Interfaceprozesse, z.B. ein auf einem Server installierter BidCoS-Service für ein HomeMatic- Lan-Interface
- die zum Ansprechen der XML-RPC-Geräte und Kanäle notwendige Zuordnungsinformation zu Interfaceprozessen und Geräteadressen.

Die Gerätezuordnung teilt sich auf zwei Dateien auf. Die Zuordnungsinformation für die fest vorhandenen Geräte (im Allgemeinen nur das SYSTEM-Gerät) befindet sich zusammen mit der Gerätebeschreibung in der Datei `system.conf`. Für die XML-RPC-Geräte, deren Interface und Adresse erst zur Laufzeit durch Konfiguration bekannt werden, befindet sich die Zuordnungsinformation in dieser Datei.

Diese Datei wird vom Logik-Switch bei Änderungen an der Gerätezuordnung automatisch geschrieben. Daher muss diese Datei vom Anwender nicht erstellt werden. Die nachfolgende Information dient nur zum besseren Verständnis.

2.3.1 Root-Element **<icserver_user>**

Das Rootelement der Systemkonfiguration ist immer **<icserver_system>**.

2.3.1.1 Untergeordnete Elemente des **<icserver_user>**-Elements

- `devices`
Dieses Element enthält eine Liste von Zuordnungsinformationen für XML-RPC-Geräte. Siehe 2.3.3.
- `interfaces`
Dieses Element enthält eine Liste der benutzerdefinierten an den Logik-Switch angebundenen XML-RPC-Interfaceprozesse. Siehe 2.2.5.4.

2.3.2 Element **<devices>**

Dieses Element enthält eine Liste von Zuordnungsinformationen für XML-RPC-Geräte in Form von **<device>**-Elementen.

2.3.2.1 Untergeordnete Elemente des **<devices>**-Elements

- `<device>`
Dieses Element beschreibt die Zuordnungsinformation für ein über XML-RPC angebundenes Gerät. Siehe 2.3.3.

2.3.3 Element <device>

Dieses Element beschreibt die Zuordnungsinformationen für ein über XML-RPC angebundenes Gerät.

2.3.3.1 Attribute des <device>-Elements

- `id [String, Pflicht]`
Dieses Attribut ist verpflichtend. Es verweist auf die Id des Gerätes, zu dem die Zuordnungsinformation gehört. Siehe 2.2.3.2.1.
Innerhalb einer Schleife (<loop>) werden in eckige Klammern eingeschlossene Ausdrücke aufgelöst.
- `interface [String, Pflicht]`
Dieses Attribut ist verpflichtend. Es verweist auf die Id des XML-RPC-Interfaceprozesses, über den das Gerät angesprochen wird.
- `address [String, Pflicht]`
Dieses Attribut ist verpflichtend. Es gibt die Adresse (Seriennummer) des Gerätes an.

2.4 Persistierung

Die Persistierungsdatei befindet sich untern der `/etc/config/icserver_values` im schreibbaren Flashbereich.

In dieser Datei speichert der Logik-Switch als Name-Werte-Paare mit Gleichheitszeichen „ini-Format“ die Werte der als persistent markierten Logiksignale. Auf das Format dieser Datei wird hier nicht weiter eingegangen.

3 XML-RPC API

Der Logik-Switch implementiert die für die Kommunikation mit den Interfaceprozessen nötigen XML-RPC-Methoden `event` und `listDevices`. Darüber hinaus gibt es einige XML-RPC-Methoden, mittels derer der Logik-Switch konfiguriert werden kann. Auf Basis dieser Methoden kann z.B. eine Weboberfläche zur Konfiguration implementiert werden.

Die XML-RPC-Methoden zur Konfiguration werden nachfolgend beschrieben.

3.1 XML-RPC-Methode `getInterfaces()`

Signatur

`Array<Struct> getInterfaces()`

Beschreibung

Diese Methode listet die dem Logik-Switch bekannten XML-RPC Interfaceprozesse auf.

Rückgabewert:

Es wird ein Array von Strukturen zurückgegeben. Jede Struktur entspricht einem XML-RPC Interfaceprozess. Die Struktur enthält folgende Felder:

- `String Id`
Die Id des Interfaceprozesses aus der Konfiguration des Logik-Switches, siehe 2.2.5.5.
- `String Url`
Die URL, unter der das XML-RPC-API des Interfaceprozesses angesprochen werden kann, wie in der Konfiguration des Logik-Switches spezifiziert.
- `Bool IsUserDefined`
Wenn dieses Feld `true` ist, wurde des Interfaceprozess in der Benutzerkonfiguration definiert (siehe 2.3).

3.2 XML-RPC-Methode *getDevices()*

Signatur

`Array<Struct> getDevices()`

Beschreibung

Diese Methode listet die dem Logik-Switch bekannten logischen Geräte auf.

Rückgabewert:

Es wird ein Array von Strukturen zurückgegeben. Jede Struktur entspricht einem logischen Gerät. Die Struktur enthält folgende Felder:

- `String Id`
Die Id des logischen Gerätes aus der Konfiguration des Logik-Switches, siehe 2.2.3.2.
- `String Type`
Der HomeMatic Gerätetyp wie in der Konfiguration des Logik-Switches spezifiziert, siehe 2.2.3.2.
- `String DisplayName`
Der Anzeigename des Gerätes wie in der Konfiguration des Logik-Switches spezifiziert, siehe 2.2.3.2.
- `String MappedInterface`
Gibt die Id des XML-RPC Interfaceprozesses für das zugeordnete physikalische Gerät an. Dieses Feld ist nur vorhanden, wenn eine Gerätezuordnung existiert.
- `String MappedAddress`
Gibt die Adresse (Seriennummer) des zugeordneten physikalischen Gerätes an. Dieses Feld ist nur vorhanden, wenn eine Gerätezuordnung existiert.
- `Bool MappedByUser`
Dieses Feld gibt an, ob die Gerätezuordnung vom Benutzer durchgeführt wurde. Wurde die Zuordnung bereits in der Systemkonfiguration durchgeführt, ist dieses Feld `false`. Nur eine vom Benutzer durchgeführte Zuordnung kann auch vom Benutzer gelöscht werden. Dieses Feld ist nur vorhanden, wenn eine Gerätezuordnung existiert.

3.3 XML-RPC-Methode *listValueConnections()*

Signatur

`Array<Struct> listValueConnections()`

Beschreibung

Diese Methode listet die Zuordnungen auf der Ebene logischer Signale auf. Aus den zurückgegebenen Informationen kann von einer Oberfläche z.B. ein Verdrahtungsplan erzeugt werden.

Rückgabewert:

Es wird ein Array von Strukturen zurückgegeben. Jede Struktur entspricht einem logischen Signal und dessen Zuordnungsinformation.

Die in der Struktur enthaltenen Felder lassen sich in Teilbereiche aufteilen. Diese Aufteilung dient nur dem besseren Verständnis.

Felder des Logiksignals:

Diese Felder entsprechen den in der Konfiguration des Logiksignals angegebenen Werten, siehe 2.2.5.3.

- `String Name`
Der Name des Logiksignals.

- `String Alias`
Der Aliasname des Logiksignals.
- `String DisplayName`
Der Anzeigename des Logiksignals.

Felder des logischen Gerätes:

Diese Felder entsprechen den in der Konfiguration des logischen Gerätes angegebenen Werten, siehe 2.2.3.2.

- `String Device`
Die Id des logischen Gerätes.
- `String DeviceDisplayName`
Der Anzeigename des logischen Gerätes.
- `Int ChannelIndex`
Die HomeMatic-Kanalnummer innerhalb des logischen Gerätes.
- `String Terminal`
Der physikalische Anschluss innerhalb des logischen Gerätes.
- `String Direction`
Gibt an, ob es sich um einen Eingang oder Ausgang handelt. Kann die Werte „Input“ oder „Output“ annehmen.

Felder des zugeordneten physikalischen Gerätekanals:

Diese Felder enthalten Informationen zum zugeordneten Kanal eines physikalischen Gerätes. Diese Felder sind nur vorhanden, wenn eine Gerätezuordnung für das Logiksignal existiert.

- `String XmlRpcInterfaceId`
Die Id des Interfaceprozesses, über den das zugeordnete Gerät angesprochen werden kann. Siehe 2.2.5.5.
- `String XmlRpcInterfaceUrl`
Die URL des Interfaceprozesses, über den das zugeordnete Gerät angesprochen werden kann. Siehe 2.2.5.5.
- `String XmlRpcDevice`
Die Adresse (Seriennummer) des zugeordneten Gerätes.
- `String XmlRpcChannel`
Die Adresse des zugeordneten Gerätekanals.
- `String XmlRpcValue`
Die Id des zugeordneten Kanalwertes.

3.4 XML-RPC-Methode *mapDevice()*

Signatur

```
void mapDevice(String deviceId, String interfaceId, String
deviceIdAddress)
```

Beschreibung

Diese Methode stellt eine neue Zuordnung zwischen einem logischen Gerät und einem physikalischen Gerät her.

Parameter:

- `String deviceId`
Die Id des logischen Gerätes aus der Konfiguration des Logik-Switches, siehe 2.2.3.2.
- `String interfaceId`
Die Id des Interfaceprozesses, über den das physikalische Gerät angesprochen wird. Aus der Konfiguration des Logik-Switches, siehe 2.2.5.5.

- String deviceAddress
Die Adresse (Seriennummer) des zuzuordnenden Gerätes.

3.5 XML-RPC-Methode *unmapDevice()*

Signatur

void unmapDevice(String deviceId)

Beschreibung

Diese Methode hebt eine Zuordnung zwischen einem logischen Gerät und einem physikalischen Gerät auf.

Parameter:

- String deviceId
Die Id des logischen Gerätes aus der Konfiguration des Logik-Switches, siehe 2.2.3.2.