



Via Netzwerk auf HomeMatic zugreifen – XML-RPC-Schnittstelle/HomeMatic®

Mittels XML-RPC-Schnittstelle kann über ein Netzwerk auf HomeMatic-Geräte zugegriffen werden. Damit lässt sich die Hausautomation in eigene Anwendungen integrieren. Dieser Artikel beschreibt die Grundlagen im Umgang mit dieser Schnittstelle.

Die HomeMatic®-Zentrale

Die HomeMatic-Zentrale stellt zur Steuerung von Aktoren und Sensoren eine Kommunikationsschnittstelle bereit, die auf XML-RPC basiert. Damit ist es möglich, Funktionen der Hausautomation in eigene Anwendungen zu integrieren. Dieser Artikel beschreibt die Grundlagen und zeigt anhand zweier Beispiele die Integration von HomeMatic in eigene Anwendungen. Die Spezifikation dieser Schnittstelle wurde von eQ-3 veröffentlicht und steht jetzt auf der HomeMatic-Homepage zur Verfügung.

Grundlagen

Bevor die Verwendung der XML-RPC-Schnittstelle anhand von Beispielen demonstriert wird, soll beschrieben werden, was mit dieser Schnittstelle möglich ist und wie man diese einsetzen kann. Abbildung 1 zeigt den schematischen Aufbau der HomeMatic-Zentrale. Im unteren Bereich sind die drei Schnittstellen zu sehen, über die HomeMatic-Geräte

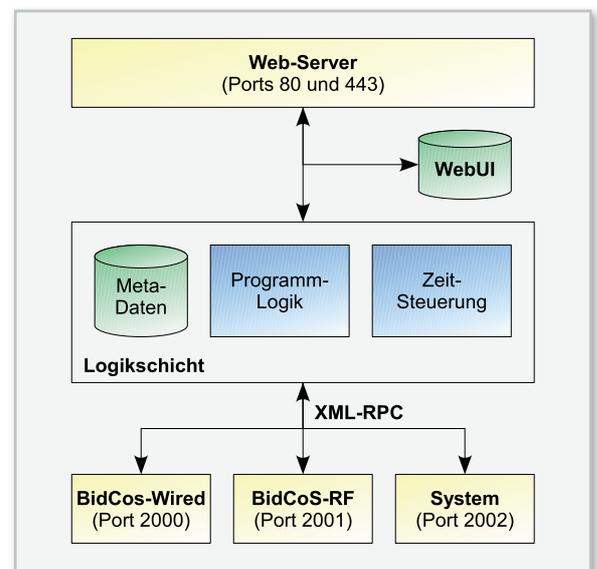


Bild 1: Schematischer Aufbau der HomeMatic-Zentrale

angeschlossen sein können. Jede dieser Schnittstellen kann über XML-RPC angesprochen werden. So lassen sich draht-

gebundene HomeMatic-Geräte über Port 2000 erreichen, funkbasierte Komponenten über Port 2001 und die internen Geräte der HomeMatic-Zentrale über Port 2002.

Die nächste Software-Schicht innerhalb der HomeMatic-Zentrale ist die sogenannte Logikschicht. Hier werden die Programmlogik und die Zeitsteuerung ausgeführt. Außerdem verwaltet die Logikschicht Metadaten wie die Namen von Geräten, Räume, Gewerke sowie die Favoriten.

Über XML-RPC lassen sich Aktoren steuern und die Zustände von Sensoren überwachen. Einen Zugriff auf die Logikschicht erlaubt diese Schnittstelle dagegen nicht.

BidCoS-Service

Neben der HomeMatic-Zentrale gibt es noch ein zweites Einsatzgebiet der XML-RPC Schnittstelle. Bei dem BidCoS-Service handelt es sich um einen Windows-Dienst, der die XML-RPC-Schnittstelle auf einem Windows-PC bereitstellt. Dadurch lassen sich HomeMatic-Geräte auch über die HomeMatic-Konfigurations-Adapter HM-CFG-LAN bzw. HM-CFG-USB ansprechen.

Der BidCoS-Service ist Bestandteil der User-Software zum HomeMatic-Konfigurations-Adapter und kann von der Home-

```
Listen Port = 2001

Log Level = 6
Persist Keys = 1

Device Description Dir = ${Bindir}\devicetypes
Device Files Dir = ${CommonAppData}\devices
Key File = ${CommonAppData}\keys
Address File = ${CommonAppData}\ids

Log Destination = File
Log Filename = ${CommonAppData}\log\bidcos.log

[Interface 0]
Type = Lan Interface
Serial Number = <Seriennummer>
Encryption Key = <Zugriffscod>
Description = <Beschreibung>
#IP Address = <IP-Adresse >

[Interface 1]
Type = Lan Interface
Serial Number = <Seriennummer>
Encryption Key = <Zugriffscod>
Description = <Beschreibung>
#IP Address = <IP-Adresse >
```

Bild 2: Konfigurationsdatei des BidCoS-Service

Matic-Homepage in der Rubrik Updates heruntergeladen werden.

Bevor der BidCoS-Service verwendet werden kann, muss dieser konfiguriert werden. Das geschieht über die Datei bidcos.conf, die sich nach erfolgreicher Installation im Anwendungsdaten-Verzeichnis befindet. Unter Windows XP ist das meist „C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Bidcos-Service“, bei Windows Vista und Windows 7 ist dies „C:\ProgramData\BidCoS-Service“. Abbildung 2 zeigt den Aufbau der Datei bidcos.conf. Der Parameter „Listen Port“ legt fest, über welchen Port der BidCoS-Service die XML-RPC-Schnittstelle bereitstellt. Üblicherweise ist hier derselbe Port wie das BidCoS-RF-Interface auf der HomeMatic-Zentrale eingestellt.

Für jeden HomeMatic-Konfigurations-Adapter, der vom BidCoS-Service verwaltet werden soll, existiert in der Konfigurationsdatei ein Interface-Abschnitt. Dabei sind diese Abschnitte von 0 beginnend durchnummeriert. Sie enthalten die Seriennummer, den Zugriffscode, eine kurze Beschreibung sowie die IP-Adresse des jeweiligen HomeMatic-Konfigurations-Adapters. Dabei ist die Angabe der IP-Adresse optional und kann weggelassen werden.

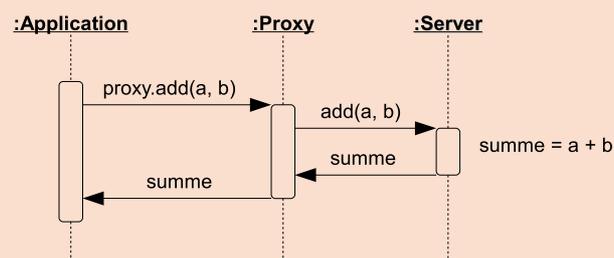
Beispiel-Programme

Im Folgenden wird die Verwendung der XML-RPC-Schnittstelle anhand zweier Beispiele demonstriert. Als Plattform für die Implementation wurde dabei .Net gewählt. Beide Beispiele sind komplett in C# geschrieben. Sie stehen jeweils als Visual-Studio-2008-Projekt mit vollständigem Quelltext im Downloadbereich zur Verfügung.

Als XML-RPC-Bibliothek wurde XML-RPC.NET von Charles Cook gewählt. Diese Bibliothek steht unter der MIT-X11-Lizenz und darf somit auch in kommerziellen Projekten verwendet werden. Mit dieser Bibliothek können XML-RPC-Clients besonders elegant implementiert werden. Wie später gezeigt wird, ist für die Generierung eines Proxy-Objekts lediglich die Angabe einer Schnittstelle nötig. Schließlich zeichnet sich XML-RPC.NET durch eine sehr gute Dokumentation aus.

Eine ausführliche Liste verfügbarer XML-RPC-Bibliotheken für eine Vielzahl anderer Programmiersprachen ist auf der XML-RPC-Homepage zu finden.

Exkurs – XML-RPC



Beispiel für einen entfernten Prozeduraufruf über einen Proxy

Bei XML-RPC handelt es sich um ein Protokoll, mit dem entfernte Prozeduraufrufe (RPC: Remote Procedure Call) durchgeführt werden. Als Transport-Protokoll setzt XML-RPC dabei auf HTTP (HyperText Transfer Protocol). Damit wird XML-RPC neben SOAP und JSON-RPC vor allem in Web-Services eingesetzt. Für praktisch alle namhaften Programmiersprachen existieren XML-RPC-Bibliotheken. Zum Aufruf der entfernten Prozeduren kommen oft Proxy-Klassen zum Einsatz. Dabei übernimmt der Proxy die gesamte XML-RPC-Logik.

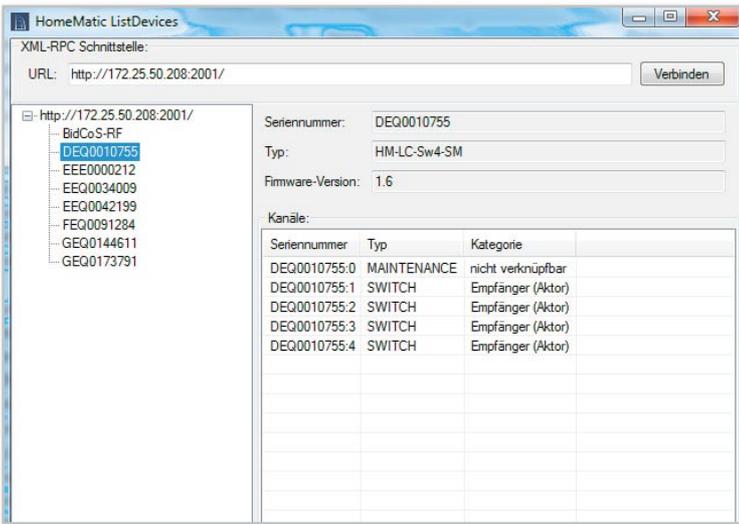


Bild 3: Beispielprogramm zum Auflisten der HomeMatic-Geräte

HomeMatic Geräte auflisten

In diesem Beispiel geht es darum, alle verfügbaren HomeMatic-Geräte aufzulisten (Abbildung 3).

Der vollständige Quelltext des Programms steht im Downloadbereich zur Verfügung. Nachdem die URL zur HomeMatic-Zentrale oder zum BidCoS-Service eingegeben wurde und der „Verbinden“-Button aktiviert wird, listet die Anwendung alle verfügbaren HomeMatic-Geräte in einer baumartigen Ansicht auf. Wählt der Anwender ein Gerät aus, werden ihm in der Detailansicht nähere Informationen über das Gerät und seine Kanäle angezeigt. Dazu stellt die XML-RPC-Schnittstelle die Methode `listDevices` zur Verfügung:

```
DeviceDescription[] listDevices();
```

```
namespace ListDevices
{
    using System;
    using System.Collections.Generic;
    using System.Text;

    using CookComputing.XmlRpc;

    public interface IHomeMaticProxy : IXmlRpcProxy
    {
        [XmlRpcMethod("listDevices")]
        DeviceDescription[] ListDevices();
    }
}
```

Bild 4: Signatur der XML-RPC-Methode `listDevices`

```
IHomeMaticProxy proxy = XmlRpcProxyGen.Create<IHomeMaticProxy>();
proxy.Timeout = 15000;
proxy.Url = "http://127.0.0.1:2001/";
DeviceDescription[] devices = proxy.ListDevices();
```

Bild 6: Proxy-Objekt instanziiieren und Geräteliste abrufen

Die Methode `listDevices` liefert ein Array von Gerätebeschreibungen. Dabei unterscheidet die XML-RPC-Schnittstelle nicht zwischen Geräten und Kanälen, wie es beispielsweise innerhalb der HomeMatic-Web-UI der Fall ist. Innerhalb der XML-RPC-Schnittstelle wird daher von logischen Geräten gesprochen, die entweder physische Geräte oder Kanäle sein können.

Eine Gerätebeschreibung wird durch die Klasse `DeviceDescription` abgebildet. In der Beispielanwendung werden nicht alle Elemente dieser Struktur verwendet. Eine vollständige Liste aller Strukturelemente ist in der Spezifikation der XML-RPC-Schnittstelle zu finden. Die wichtigsten Elemente in der `DeviceDescription` sind „ADDRESS“ und „TYPE“. Dabei enthält „ADDRESS“ die Seriennummer des logischen Geräts. Dieser Wert wird bei jeder Kommunikation benötigt. Das Element „TYPE“ enthält für physische Geräte deren Kurzbezeichnung und für Kanäle den Kanaltyp.

Damit in der Beispielanwendung die Kommunikation mit der XML-RPC-Schnittstelle realisiert werden kann, wird ein geeigneter Proxy benötigt. Wie zuvor erwähnt, reicht hierfür die Definition einer Schnittstelle aus. Diese ist in Abbildung 4 dargestellt.

In der baumartigen Ansicht werden nur physische Geräte dargestellt. Kanäle sollen dort nicht zu finden sein, sondern nur in der Detailansicht auftauchen. Um die Unterscheidung zwischen Gerät und Kanal treffen zu können, wird das Element „PARENT“ verwendet (Abbildung 5). Dies gehört zu den obligatorischen Elementen, die in jeder `DeviceDescription` enthalten sein müssen. Bei Kanälen enthält „PARENT“ die Seriennummer des übergeordneten Geräts, bei Geräten ist das Element jedoch leer.

Abbildung 6 zeigt die Verwendung des Proxys und das Auflisten der HomeMatic-Geräte. Im Gegensatz zu Abbildung 6 verwendet die Beispielanwendung ein Text-Eingabefeld für die URL des Proxys.

```
namespace ListDevices
{
    using System;
    using System.Collections.Generic;
    using System.Text;

    using CookComputing.XmlRpc;

    public class DeviceDescription
    {
        [XmlRpcMember("TYPE")]
        public string Type;

        [XmlRpcMember("ADDRESS")]
        public string Address;

        [XmlRpcMember("PARENT")]
        public string Parent;

        [XmlRpcMissingMapping(MappingAction.Ignore), XmlRpcMember("CHILDREN")]
        public string[] Children;

        [XmlRpcMissingMapping(MappingAction.Ignore), XmlRpcMember("PARENT_TYPE")]
        public string ParentType;

        [XmlRpcMissingMapping(MappingAction.Ignore), XmlRpcMember("FIRMWARE")]
        public string Firmware;

        [XmlRpcMissingMapping(MappingAction.Ignore), XmlRpcMember("DIRECTION")]
        public int Direction;

        public bool IsDevice()
        {
            return string.IsNullOrEmpty(this.Parent);
        }

        public bool IsChannel()
        {
            return !this.IsDevice();
        }

        public override string ToString()
        {
            return this.Address;
        }

        public string GetCategory()
        {
            switch (this.Direction)
            {
                case 0:
                    return "nicht verknüpfbar";
                case 1:
                    return "Sender (Sensor)";
                case 2:
                    return "Empfänger (Aktor)";
                default:
                    return string.Empty;
            }
        }
    }
}
```

Bild 5: Die Klasse DeviceDescription

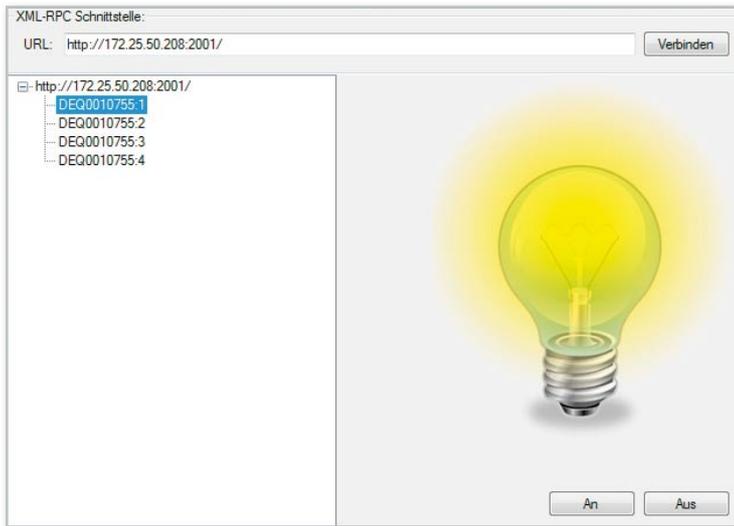


Bild 7: HomeMatic-Geräte schalten

HomeMatic-Geräte schalten

In diesem Beispiel geht es darum, HomeMatic-Geräte zu schalten und deren Zustand abzufragen (Abbildung 7). Wie im vorangegangenen Beispiel muss zunächst die URL zur HomeMatic-Zentrale bzw. dem BidCoS-Service eingegeben werden. Nach dem Verbinden werden jedoch nicht alle Geräte in der baumartigen Ansicht dargestellt. Lediglich schaltbare Kanäle sind dort zu finden. Wählt der Anwender einen dieser Kanäle aus, wird dessen Zustand durch eine Glühbirne symbolisiert. Er hat nun die Möglichkeit, den ausgewählten Kanal zu schalten. Zusätzlich wird der aktuelle Zustand des ausgewählten Kanals mit Hilfe eines Intervall-Timers überwacht.

Im Gegensatz zur Auflistung der HomeMatic-Geräte werden hier ausschließlich schaltbare Kanäle dargestellt. Um diese zu erkennen, werden die Elemente „TYPE“ sowie „PARENT_TYPE“ ausgewertet. Das Element „PARENT_TYPE“ enthält die Kurzbezeichnung des übergeordneten HomeMatic-Geräts. Bei Schaltaktoren ist das zum Beispiel „HM-LC-Sw1-PI“ oder „HM-LC-Sw4-PI“. Über das Element „TYPE“ werden nun die schaltbaren Kanäle identifiziert. Diese sind vom Typ „SWITCH“.

Im nächsten Schritt wird das IHomeMaticProxy-Interface um die Methoden GetValue und SetValue erweitert (siehe Abbildung 8). Mit diesen Methoden können Datenpunkte eines Kanals gelesen bzw. beschrieben werden. Eine vollständige Auflistung aller Datenpunkte kann auf der HomeMatic-Homepage heruntergeladen werden.

Der Zustand des HomeMatic-Schaltaktors verbirgt sich hinter dem Datenpunkt „STATE“. Abbildung 9 demonstriert, wie der Zustand eines solchen Schaltaktors gelesen bzw. geschrieben wird. Dabei ist „DEQ0010755:1“ die Seriennummer des betreffenden Kanals.

Fazit

In diesem Artikel wurde die XML-RPC-Schnittstelle zur Steuerung von HomeMatic-Geräten vorgestellt. Es wurde gezeigt, wie mit Hilfe dieser Schnittstelle einfache Aufgaben, wie das Auflisten sowie das Schalten von Geräten, realisiert werden können. Hiermit stehen dem versierten Anwender viele Möglichkeiten offen, um HomeMatic in eigene Anwendungen zu integrieren. 

```
namespace Switch
{
    using System;
    using System.Collections.Generic;
    using System.Text;

    using CookComputing.XmlRpc;

    public interface IHomeMaticProxy : IXmlRpcProxy
    {
        [XmlRpcMethod("listDevices")]
        DeviceDescription[] ListDevices();

        [XmlRpcMethod("setValue")]
        void SetValue(string address, string valueKey, object value);

        [XmlRpcMethod("getValue")]
        object GetValue(string address, string valueKey);
    }
}
```

Bild 8: Die Schnittstelle IHomeMaticProxy, erweitert um die Methoden GetValue und SetValue

```
IHomeMaticProxy proxy = XmlRpcProxyGen.Create<IHomeMaticProxy>();
proxy.Timeout = 15000;
proxy.Url = "http://127.0.0.1:2001/";

// einschalten
proxy.SetValue("DEQ0010755:1", "STATE", true);

// ausschalten
proxy.SetValue("DEQ0010755:1", "STATE", false);

// Zustand ermitteln
bool state = (bool) proxy.GetValue("DEQ0010755:1", "STATE");
```

Bild 9: Funk-Schaltaktor schalten

Links:

HomeMatic-Homepage

www.homematic.com

Usersoftware zum HomeMatic-Konfigurations-Adapter

www.homematic.com/index.php?id=644&L=0

Technische Dokumentationen zu HomeMatic

www.homematic.com/index.php?id=156

XML-RPC-Homepage www.xmlrpc.com

Liste verschiedener XML-RPC-Bibliotheken

www.xmlrpc.com/directory/1568/implementations

XML-RPC.NET-Bibliothek www.xml-rpc.net

Projekte im Downloadbereich

www.elv.de/output/controller.aspx?cid=588