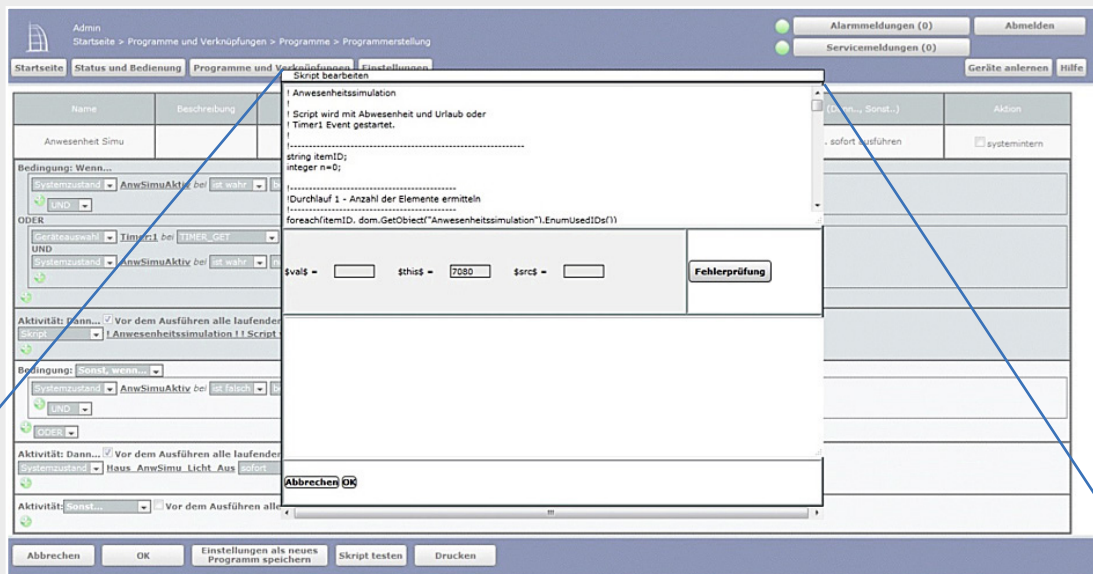




Homematic Scriptprogrammierung

Teil 2 – Objekte, Datenpunkte und erste Scripts



Script bearbeiten

```
foreach(itemID, dom.GetObject("Anwesenheitssimulation").EnumUsedIDs())
{
  var item = dom.GetObject(itemID);
  var device = dom.GetObject(item.Device());
  if (item.IsTypeOf(OT_CHANNEL))
  {
    if (device.HssType().Find("HM-LC-Sw")>=0)
    {
      n=n+1;
    }
    if (device.HssType().Find("HM-LC-Dim")>=0)
    {
      n=n+1;
    }
  }
}
```

\$val\$ = \$this\$ = \$src\$ = **Fehlerprüfung**

Im zweiten Teil der Artikelserie über die Scriptprogrammierung beschäftigen wir uns mit Objekten und Datenpunkten und wir schreiben die ersten Scripts.

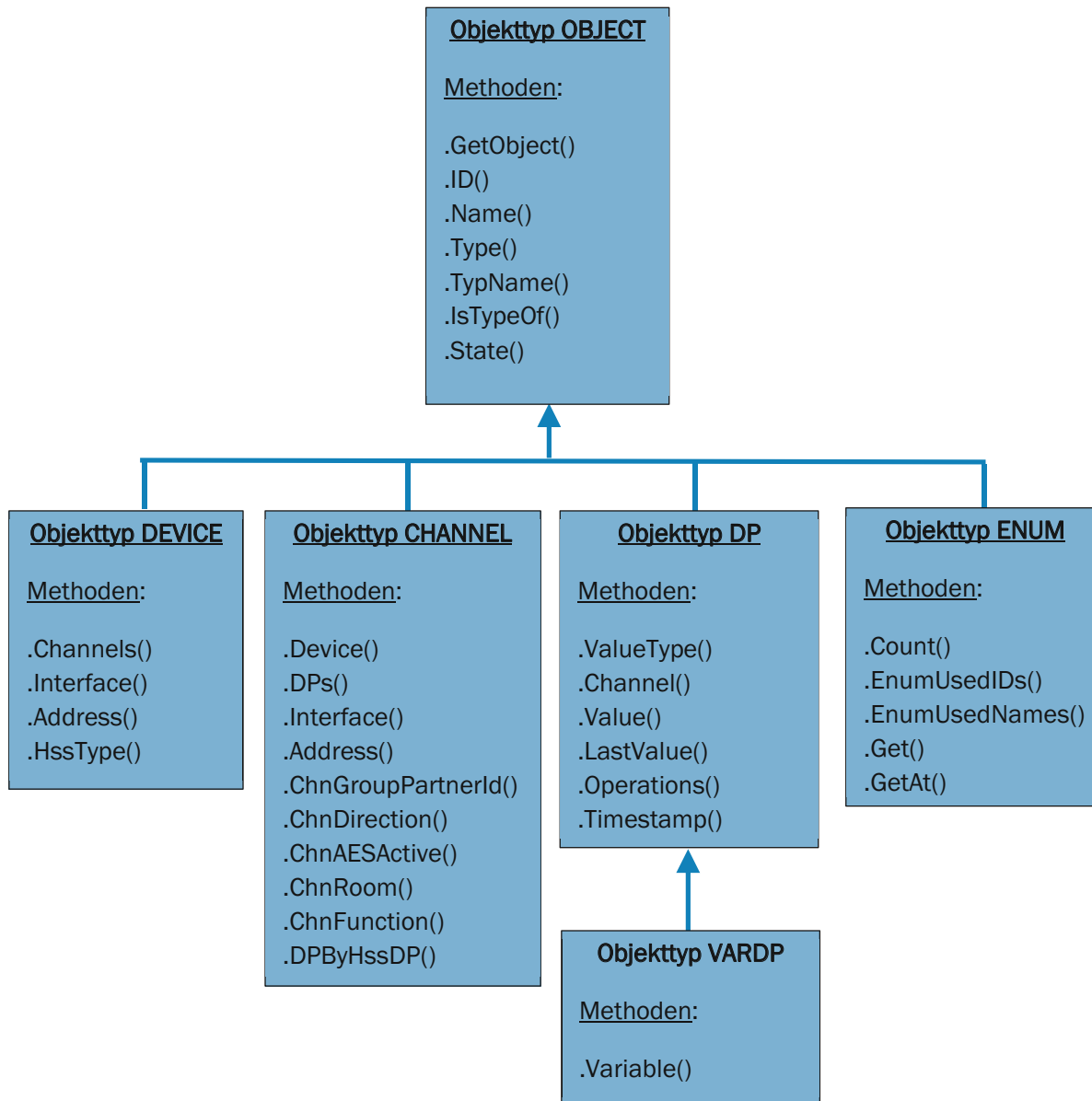


Objekte

Das Homematic Objektmodell

Um den Zustand von Homematic Initiatoren und Aktoren oder Systemvariablen per Scriptprogrammierung abzufragen bzw. zu schalten oder zu manipulieren, bedient man sich des Homematic Objektmodells.

Dieses beschreibt die Objekttypen des Homematic Systems und die Methoden, mit deren Hilfe das Abfragen bzw. Schalten und Manipulieren möglich ist:





In den folgenden Beispielen zur Verdeutlichung der Methoden gehen wir von einem Objekt „Dimmer Wohnzimmer“, einem Unterputz-Dimmer für Markenschalter, aus:

Admin											
Startseite > Einstellungen > Geräte											
Startseite Status und Bedienung Programme und Verknüpfungen Einstellungen											
Dimmer Wohnzimmer	HM-LC-Dim1TPBU-FM		Funk-Dimmaktor 1-fach für Markenschalter, Phasenabschnitt, Unterputzmontage	JEQ013727	BidCos-RF	Standard	Anwesenheitssimulation Licht Siri	Wohnzimmer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Einstellen Löschen Direkte Programme
Dimmer Wohnzimmer:1	HM-LC-Dim1TPBU-FM		Funk-Dimmaktor 1-fach für Markenschalter, Phasenabschnitt, Unterputzmontage	JEQ013727:1	Empfänger	Standard	Anwesenheitssimulation Licht Siri	Wohnzimmer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Einstellen Direkte Programme
Dimmer Wohnzimmer:2	HM-LC-Dim1TPBU-FM		Funk-Dimmaktor 1-fach für Markenschalter, Phasenabschnitt, Unterputzmontage	JEQ013727:2	Empfänger	Standard	Licht		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Einstellen Programme
Dimmer Wohnzimmer:3	HM-LC-Dim1TPBU-FM		Funk-Dimmaktor 1-fach für Markenschalter, Phasenabschnitt, Unterputzmontage	JEQ013727:3	Empfänger	Standard	Licht		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Einstellen Direkte Programme

Grundsätzlich funktioniert die zentrale Methode „GetObject“ mit jedem Objekt. Da sich häufig kein Objekt im Zugriff befindet, wird in der Praxis das globale Objekt „dom“ verwendet, um auf Objekte zuzugreifen:

```
var myObject = dom.GetObject("Dimmer Wohnzimmer");
```

Der Objekttyp „Objekt“

Name Prototyp	Kurzbeschreibung	Beispiel
GetObject var object.GetObject(integer id) var object.GetObject(string name)	liefert ein Objekt anhand seiner ID bzw. seines Namens	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer"); =>myObj hat in diesem Beispiel den Wert "Dimmer Wohnzimmer"</pre>
ID integer object.ID()	liefert die ID eines Objekts	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer").ID(); =>myObj hat in diesem Beispiel den Wert 13727</pre>
Name string object.Name()	liefert den Namen eines Objekts	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer").Name(); =>myObj hat in diesem Beispiel den Wert "Dimmer Wohnzimmer"</pre>
TypeName string object.TypeName()	liefert die Bezeichnung des Objekttyps	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer").TypeName(); =>myObj hat in diesem Beispiel den Wert "DEVICE"</pre>
IsTypeOf boolean object.IsTypeOf(integer typeld)	prüft, ob ein Objekt einen speziellen Typ implementiert mögliche Typen: OT_OBJECT=Objekt OT_ENUM=Aufzählung OT_DEVICE=Gerät OT_CHANNEL=Kanal OT_DP=Datenpunkt OT_VARDP=Systemvariable	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer").IsTypeOf(OT_DEVICE); =>myObj hat in diesem Beispiel den Wert true string sName="Dimmer Wohnzimmer"; var myObj=dom.GetObject(sName).IsTypeOf(OT_CHANNEL); =>myObj hat in diesem Beispiel den Wert false</pre>
State var object.State() boolean object.State(boolean newState) boolean object.State(integer newState) boolean object.State(real newState) boolean object.State(time newState) boolean object.State(string newState)	ermittelt oder setzt den Zustand eines Objekts	<pre>var myObj =dom.GetObject("Dimmer Wohnzimmer").State(); =>myObj hat in diesem Beispiel den Wert "", also "nichts". Der Grund: Der Wert des Dimmers (0.00 ... 1.00) steht im Kanal 1 des Dimmerobjekts (siehe Homematic Script Dokumentation Teil 4: Datenpunkte) Folglich muss es also richtig heißen: var myObj =dom.GetObject("Dimmer Wohnzimmer:1").State(); =>myObj hat in diesem Beispiel nun den Wert: 0,00 - wenn der Dimmer ausgeschaltet ist, 1,00 - wenn der Dimmer auf 100 % eingeschaltet ist, 0,50 - wenn der Dimmer auf 50 % eingeschaltet ist.</pre>



Der Objekttyp „Device“

Name	Prototyp	Kurzbeschreibung	Beispiel
Channels	object.device.Channels()	liefert die Liste der Kanäle in dem Gerät	<pre>string sName = "Dimmer Wohnzimmer"; var myObj=dom.GetObject(sName).Channels();</pre> myObj hat in diesem Beispiel den Wert "13745\t13765\t13785" <u>Anmerkung:</u> \t sind – wie im Teil 1 beschrieben, die Listentrenner! <pre>string sName = "Dimmer Wohnzimmer"; var myObj=dom.GetObjects(Name).Device().Channels().EnumUsedNames();</pre> myObj hat in diesem Beispiel den Wert "Dimmer Wohnzimmer:1\tDimmer Wohnzimmer:2\tDimmer Wohnzimmer:3"
Interface	object.device.Interface()	liefert die ID der Schnittstelle, an der das Gerät angeschlossen ist zzt. 3 mögliche Schnittstellen: <ul style="list-style-type: none"> • BidCos-RF für Funkkomponenten • BidCos-Wired für drahtgebundene Geräte • System für interne Geräte 	<pre>var myObj =dom.GetObject("Dimmer Wohnzimmer").Interface();</pre> myObj hat in diesem Beispiel den Wert 1007
Address	object.device.Address()	liefert die Seriennummer des Geräts	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer").Address();</pre> myObj hat in diesem Beispiel den Wert "JEQxxxxxxx"
HssType	object.device.HssType()	liefert die Kurzbezeichnung des Homematic Gerätetyps	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer").HssType ();</pre> myObj hat in diesem Beispiel den Wert "HM-LC-Dim1TPBU-FM"

Der Objekttyp „Channels“ beschreibt die Kanäle eines Device.

Name	Prototyp	Kurzbeschreibung	Beispiel
Device	integer channel.Device()	liefert die ID des Geräts, in dem der Kanal definiert ist	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer:1").Device();</pre> myObj hat in diesem Beispiel den Wert "Dimmer Wohnzimmer".
DPs	object channel.DPs()	liefert eine Liste der Datenpunkte des Kanals	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer:1").DPs();</pre> myObj hat in diesem Beispiel den Wert: "13760\t13761\t13763\t13762\t13758\t13754\t13750\t13746\t17678\t34750" Die Namen (Bezeichnungen) der Kanäle erhält man mit: <pre>string sName = "Dimmer Wohnzimmer:1"; var myObj =dom.GetObject(sName).DPs().EnumUsedNames();</pre> myObj hat in diesem Beispiel den Wert: "BidCos-RF.JEQxxxxxxx:1.LEVEL\t BidCos-RF.JEQxxxxxxx:1.OLD_LEVEL\t BidCos-RF.JEQxxxxxxx:1.RAMP_TIME\t BidCos-RF.JEQxxxxxxx:1.ON_TIME\t BidCos-RF.JEQxxxxxxx:1.INHIBIT\t BidCos-RF.JEQxxxxxxx:1.ERROR_REDUCED\t BidCos-RF.JEQxxxxxxx:1.ERROR_OVERLOAD\t BidCos-RF.JEQxxxxxxx:1.ERROR_OVERHEAT\t BidCos-RF.JEQxxxxxxx:1.RAMP_STOP\t BidCos-RF.JEQxxxxxxx:1.LEVEL_REAL" Hinweis: Eine Auflistung aller Kanäle aller Devices finden Sie in der Dokumentation <i>Homematic Script Dokumentation Teil 4: Datenpunkte</i>
Interface	integer channel.Interface()	liefert die ID der Schnittstelle, über die der Kanal angeschlossen ist	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer:1").Interface();</pre> myObj hat in diesem Beispiel den Wert 1007.
Address	string channel.Address()	liefert die Seriennummer des Kanals	<pre>var myObj=dom.GetObject("Dimmer Wohnzimmer:1").Address();</pre> myObj hat in diesem Beispiel den Wert "JEQxxxxxxx:1".
ChnGroupPartnerId	integer channel.ChnGroupPartnerId()	liefert die ID des Partners in einer Kanalgruppe	<pre>string sName="Dimmer Wohnzimmer:1"; var myObj=dom.GetObject(sName).ChnGroupPartnerId();</pre> myObj hat in diesem Beispiel den Wert 65535.
ChnDirection	integer channel.ChnDirection()	ermittelt die Kategorie des Kanals Kategorien: 0 = Kanal nicht verknüpfbar 1 = Sender (Sensor) 2 = Empfänger (Aktor)	<pre>string sName="Dimmer Wohnzimmer:1"; var myObj=dom.GetObject(sName).ChnDirection();</pre> myObj hat in diesem Beispiel den Wert 2.
ChnAESActive	boolean channel.ChnAESActive()	ermittelt, ob der Kanal AES-verschlüsselt sendet	<pre>string sName="Dimmer Wohnzimmer:1"; boolean bVerschl=dom.GetObject(sName).ChnAESActive();</pre> bVerschl hat in diesem Beispiel den Wert false.



ChnArchive	boolean channel.ChnArchive()	ermittelt, ob der Kanal protokolliert wird	<code>string sName="Dimmer Wohnzimmer:1";</code> <code>boolean bProto=dom.GetObject(sName).ChnActive();</code> bProto hat in diesem Beispiel den Wert false
ChnRoom	string channel.ChnRoom()	ermittelt die Räume, denen der Kanal zugeordnet ist	<code>string sName="Dimmer Wohnzimmer:1";</code> <code>string sRooms=dom.GetObject(sName).ChnRoom();</code> sRooms hat in diesem Beispiel den Wert "1293". Der Rückgabewert ist eine Liste von ID's (in diesem Fall nur eine ID, da nur einem Raum zugeordnet), deren Namen in einer foreach-Schleife durch die Funktion .Name() ermittelt werden kann: <code>var myChannel=dom.GetObject("Dimmer Wohnzimmer:1");</code> <code>string srooms="";</code> <code>string sld;</code> <code>foreach(sld, myChannel.ChnRoom())</code> { <code>var sroom=dom.GetObject(sld);</code> <code>srooms=srooms # sroom.Name() # " ";</code> } Die Stringvariable srooms hat danach den Wert: "Wohnzimmer"
ChnFunction	string channel.ChnFunction()	ermittelt die Gewerke, denen der Kanal zugeordnet ist	<code>string sNameKanal="Dimmer Wohnzimmer:1"</code> <code>var myObj=dom.GetObject(sNameKanal).ChnFunction();</code> myObj hat in diesem Beispiel den Wert 5192\t1253\t6493 Die Namen der Gewerke, die die 3 ID's 5192, 1253 und 6493 darstellen, erhält man mit folgendem Script: <code>string sGewerke="";</code> <code>string sld;</code> <code>foreach(sld, myChannel.ChnFunction())</code> { <code>var sGewerk=dom.GetObject(sld);</code> <code>sGewerke=sGewerke # sGewerk.Name() # " ";</code> } sGewerke hat in diesem Beispiel den Wert "Anwesenheitssimulation Licht Siri"
DPByHssDP	object channel.DPByHssDP(string name)	ermittelt einen Datenpunkt des Kanals anhand seines Namens	<code>string sNameKanal="Dimmer Wohnzimmer:1"</code> <code>var myVal=dom.GetObject(sNameKanal).DPByHssDP("LEVEL").State();</code> Je nach eingestellter Helligkeit hat die Variable myVal den Wert: 0,00 bei ausgeschalteter Lampe, 0,40 bei 40 % Helligkeit, 0,90 bei 90 % Helligkeit, 1,00 bei 100 % Helligkeit.

Der Objekttyp „Datenpunkte“

Name	Prototyp	Kurzbeschreibung	Beispiel
ValueType	integer dp.ValueType()	ermittelt den Datentyp des Wertes, den der Datenpunkt repräsentiert.	<code>string sNameKanal="Dimmer Wohnzimmer:1"</code> <code>var myVal=dom.GetObject(sNameKanal).DPByHssDP("LEVEL").ValueType();</code> Die Variable myVal hat in diesem Beispiel den Wert 6.
Channel	integer dp.Channel()	liefert die ID des Kanals, zu dem der Datenpunkt gehört	<code>string sNameKanal="Dimmer Wohnzimmer:1"</code> <code>var myVal=dom.GetObject(sNameKanal).DPByHssDP("LEVEL").Channel();</code> Die Variable myVal hat in diesem Beispiel den Wert 13745.
Value	var dp.Value()	liefert den aktuellen Wert des Datenpunktes	<code>string sNameKanal="Dimmer Wohnzimmer:1"</code> <code>var myVal=dom.GetObject(sNameKanal).DPByHssDP("LEVEL").Value();</code> Je nach eingestellter Helligkeit hat die Variable myVal den Wert: 0,00 bei ausgeschalteter Lampe 0,40 bei 40% Helligkeit, 1,00 bei 100% Helligkeit.
LastValue	var dp.LastValue()	liefert den Wert des Datenpunktes vor der letzten Aktualisierung	<code>string sNameKanal="Dimmer Wohnzimmer:1"</code> <code>var myVal=dom.GetObject(sNameKanal).DPByHssDP("LEVEL").LastValue();</code> Nach einer Werteänderung hat die Variable myVal je nach eingestellter Helligkeit vor der Änderung z.B. den Wert: 0,80 bei 80% Helligkeit vor der Änderung, 0,10 bei 10% Helligkeit vor der Änderung.
Operations	integer dp.Operations()	ermittelt, welche Operationen auf dem Datenpunkt ausgeführt werden können	<code>string sNamKan="Dimmer Wohnzimmer:1"</code> <code>var myVal=dom.GetObject(sNamKan).DPByHssDP("LEVEL").Operations();</code> Die Variable myVal hat in diesem Beispiel den Wert 7.
Timestamp	time dp.Timestamp()	Zeitstempel der letzten Aktualisierung	<code>string sNamKan="Dimmer Wohnzimmer:1"</code> <code>var myVal=dom.GetObject(sNamKan).DPByHssDP("LEVEL").Timestamp();</code> Die Variable myVal hat in diesem Beispiel den Wert "2016-12-06 09:38:44" wenn am 6.12.2016 um 9Uhr38 und 44 Sekunden die letzte Werteänderung am Aktor erfolgte.



Der Objekttyp „Systemvariable“

Name	Prototyp	Kurzbeschreibung	Im Beispiel
Variable	String dp.Variable() Boolean dp.Variable(var Value)	ermittelt oder setzt den Wert einer Systemvariablen	Legen Sie eine CCU-Systemvariable sTEST als Zeichenkette an und geben Sie ihr den Wert "Dies ist eine Testvariable" Nach dem folgenden Script: <code>var sAusgabe=dom.GetObject("sTEST").State();</code> hat sAusgabe den Wert "Dies ist eine Testvariable"

Der Objekttyp „Aufzählungen“

Name	Prototyp	Kurzbeschreibung	Im Beispiel
Count	integer enum.Count()	liefert die Anzahl der Objekte in der Aufzählung	<code>var Anz=dom.GetObject("Wohnzimmer").Count();</code> Hier (im Testsystem) hat Anz den Wert 79 – dies entspricht der Anzahl der in diesem Fall verfügbaren Kanäle im Wohnzimmer. (Der Wert variiert natürlich je nach Anzahl der eingebauten Geräte). (siehe unten EnumUsedIDs)
EnumUsedIDs	string enum.EnumUsedIDs()	liefert die IDs der Objekte in der Aufzählung	<code>var raum = dom.GetObject("Wohnzimmer");</code> <code>var chns = raum.EnumUsedIDs();</code> <code>var list = "";</code> <code>string id;</code> <code>foreach(id, chns)</code> { <code>var chn = dom.GetObject(id);</code> <code>list = list # chn.Name() # " ";</code> } list hat in diesem Beispiel den Inhalt: Anzeige_Wohnzimmer:1 Dimmer LED Sued Dimmer LED WZEZ Dimmer Wohnzimmer:1 Schaltsteckdose_Fernseher:1 Wohnzimmer*Bewegungsmelder*01:1 Wohnzimmer*Bewegungsmelder*02:1 Wohnzimmer*Fensterkontakt*01:1 ... usw.
EnumUsedNames	string enum.EnumUsedNames()	liefert die Namen der Objekte in der Aufzählung	<code>var raum = dom.GetObject("Wohnzimmer");</code> <code>var chns = raum.EnumUsedNames();</code> <code>var list = "";</code> <code>string name;</code> <code>foreach(name, chns)</code> { <code>list = list # name # " ";</code> } list hat in diesem Beispiel den Inhalt: Anzeige_Wohnzimmer:1 Dimmer LED Sued Dimmer LED WZEZ Dimmer Wohnzimmer:1 Schaltsteckdose_Fernseher:1 Wohnzimmer*Bewegungsmelder*01:1 Wohnzimmer*Bewegungsmelder*02:1 Wohnzimmer*Fensterkontakt*01:1 ... usw.
Get	object enum.Get(string name)	liefert das Objekt mit dem angegebenen Namen oder „null“, wenn nichts gefunden wurde	<code>var raum = dom.GetObject("Wohnzimmer");</code> <code>var chn = raum.Get("Dimmer Wohnzimmer:1");</code> chn hat den Wert "Dimmer Wohnzimmer:1"
GetAt	object enum.GetAt(integer index)	liefert das Objekt mit dem angegebenen Index oder Null, wenn nichts gefunden wurde	<code>var raum = dom.GetObject("Wohnzimmer");</code> <code>var chn = raum.GetAt(0);</code> chn hat den Wert "Anzeige_Wohnzimmer:1" (siehe dazu auch die Beispiele oben)

Datenpunkte

Alle Homematic Geräte besitzen eine (unterschiedliche) Anzahl von Datenpunkten. Diese Datenpunkte sind im Dokument „HomeMatic-Script Dokumentation Teil 4: Datenpunkte“ für jedes Gerät ausführlich beschrieben.

Grundsätzlich besteht ein Homematic Gerät mindestens aus einem oder aus mehreren Kanälen. Und genau über diese Kanäle wird die eigentliche Funktionalität des Geräts abgebildet.



Kanäle können grob in zwei Kategorien eingeteilt werden:

- Sendekanäle (Sensoren) – geben die Information von „Sensoren“ im Gerät weiter (z. B. Helligkeitswert bei einem Bewegungsmelder)
- Empfangskanäle (Aktoren) – führen eine Funktion aus (z. B. Einschalten des Lichts bei einem Unterputz-Schalter)

Somit können in einem Homematic Gerät Kanäle mit Sendefunktion, Kanäle mit Empfangsfunktion, aber auch beide Arten von Kanälen vorhanden sein.

Sowohl Sendekanäle als auch Empfangskanäle besitzen Zustände. Diese Zustände können – je nach Art – nur lesbar oder nur setzbar oder lesbar und setzbar sein.

Nun kann jeder Kanal eine ganze Menge sogenannter Datenpunkte besitzen. Solch ein Datenpunkt symbolisiert einen Teil der Zustände eines Kanals. Das Dokument „HomeMatic-Script Dokumentation Teil 4: Datenpunkte“ listet sämtliche unter Homematic Script verfügbaren Datenpunkte aller Homematic Geräte auf.

Ein Datenpunkt besitzt immer einen Namen und einen Datentyp. Seine Zugriffsart muss gekennzeichnet sein, und er wird durch eine Menge von Attributen gekennzeichnet.

Beispiel: Unterputz-Funk-Abschnitt-Dimmaktor für Markenschalter

Funk-Abschnitt-Dimmaktor ab V2.5

Unterstützte Gerätetypen

- HM-LC-Dim1TPBU-FM
- HM-LC-Dim1TPBU-FM-2

Kanaltypen

Kanaltyp	Kanalnummer
DIMMER	1
VIRTUAL_DIMMER	2 bis 3

Kanaltyp DIMMER

Name	Typ	Zugriff
LEVEL	float	<ul style="list-style-type: none"> • lesend • schreibend • über Ereignisse
OLD_LEVEL	action	<ul style="list-style-type: none"> • schreibend
LEVEL_REAL	float	<ul style="list-style-type: none"> • lesend • über Ereignisse
RAMP_TIME	float	<ul style="list-style-type: none"> • schreibend
ON_TIME	float	<ul style="list-style-type: none"> • schreibend
RAMP_STOP	action	<ul style="list-style-type: none"> • schreibend
INHIBIT	boolean	<ul style="list-style-type: none"> • lesend • schreibend • über Ereignisse
WORKING	boolean	<ul style="list-style-type: none"> • lesend • über Ereignisse
DIRECTION	option	<ul style="list-style-type: none"> • lesend • über Ereignisse
ERROR_REDUCED	boolean	<ul style="list-style-type: none"> • lesend • über Ereignisse
ERROR_OVERLOAD	boolean	<ul style="list-style-type: none"> • lesend • über Ereignisse
ERROR_OVERHEAT	boolean	<ul style="list-style-type: none"> • lesend • über Ereignisse
INSTALL_TEST	action	<ul style="list-style-type: none"> • schreibend

Dieser Gerätetyp besitzt 3 Kanäle, wobei die Datenpunkte des ersten Kanals die für die hier beschriebene Ansteuerung interessanten Datenpunkte sind: (LEVEL, OLD_LEVEL etc.).

Die weiteren Kanäle sind virtuelle Kanäle, auf die in einer späteren Folge dieser Reihe eingegangen wird.

Beispiele für die Ansteuerung des Dimmers:

Einschalten mit 100 %:

```
string sNamKan = "Dimmer Wohnzimmer:1";
dom.GetObject sNamKan).DPByHssDP("LEVEL").State(1.0);
```

Rampe auf 2 Sekunden und Einschalten:

```
string sNamKan = "Dimmer Wohnzimmer:1";
dom.GetObject(sNamKan).DPByHssDP("RAMP_TIME").State(2);
dom.GetObject sNamKan).DPByHssDP("LEVEL").State(1.0);
```

Rampe auf 2 Sekunden und Einschaltedauer auf 5 Sekunden und Einschalten:

```
string sNamKan = "Dimmer Wohnzimmer:1";
dom.GetObject(sNamKan).DPByHssDP("RAMP_TIME").State(2);
dom.GetObject sNamKan).DPByHssDP("ON_TIME").State(5);
dom.GetObject sNamKan).DPByHssDP("LEVEL").State(1.0);
```

Die Beschreibung des Schreibens von Scripten können Sie im Web-Shop unter Webcode #10052 kostenlos herunterladen.

Ausblick

Im dritten Teil dieser Reihe werden wir uns mit dem Namensraum-System beschäftigen sowie weitere Beispiele für Scripte zeigen. 