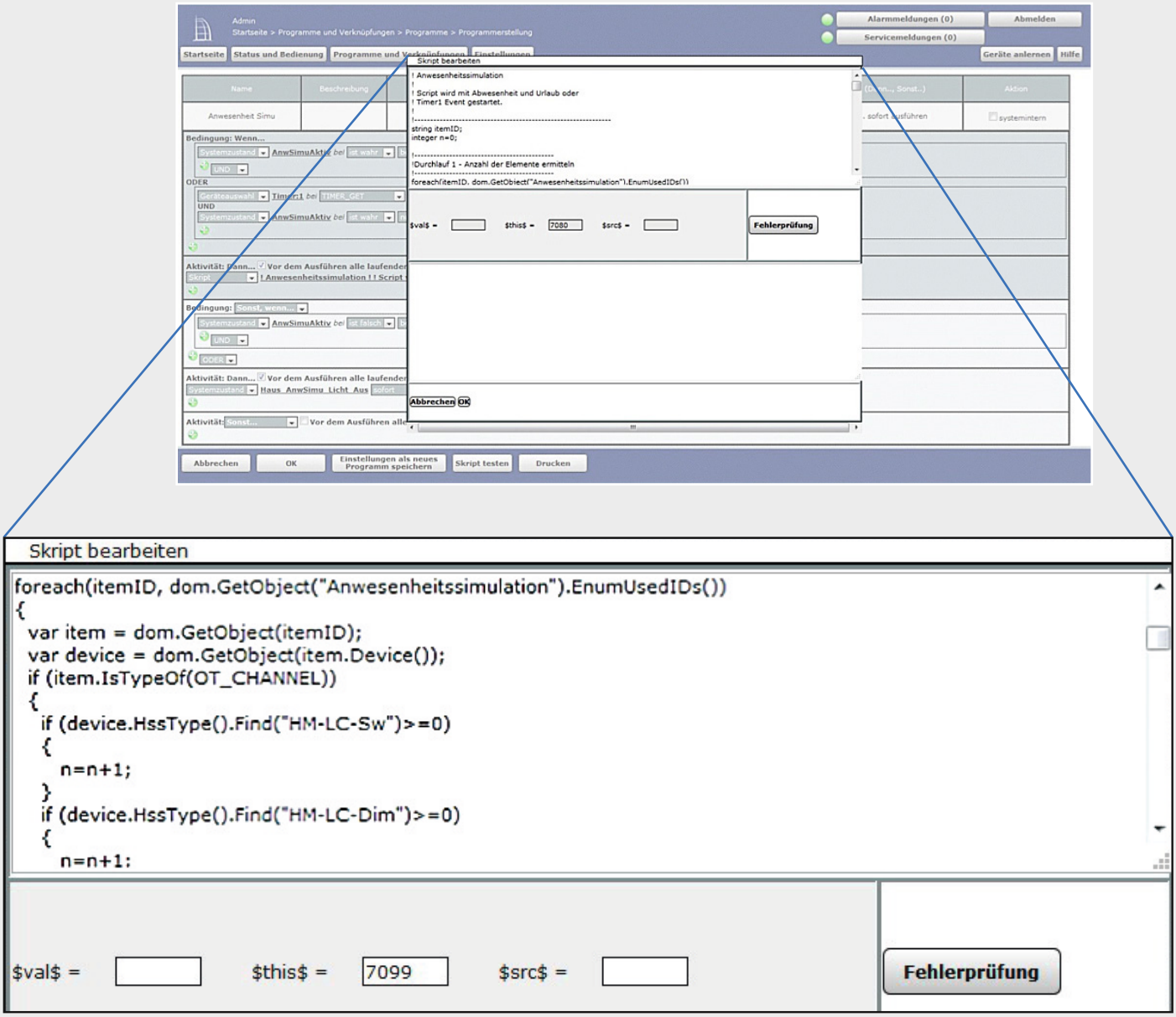
Homematic Scriptprogrammierung

Teil 3 – Namensraum „System“ und Scripte



Im dritten Teil der Artikelserie über die Scriptprogrammierung beschäftigen wir uns mit dem Namensraum „System“ und wir schreiben weitere Scripte.

Der Namensraum „System“

Im Namensraum „System“ sind allgemeine Systemfunktionen zusammengefasst. Dazu gehören Uhrzeitfunktionen oder die Behandlung von Systemvariablen.

Folgende Funktionen stehen zur Verfügung:

Name	Prototyp	Kurzbeschreibung	Beispiel
System.Date	string system.Date(string format)	Fragt die aktuelle Uhrzeit ab und stellt sie im durch (string format) vorgegebenen Format bereit. Die Platzhalter für die einzelnen Komponenten zur Bildung des Formatstrings wurden in Teil 1 der Artikelserie beschrieben.	*time t = @2008-12-24 18:30:00@;* *string sFormatiert = t.Format(“%a“);* *sFormatiert wird zu: „Wed“* *time t = @2008-12-24 18:30:00@;* *string sFormatiert = t.Format(“%c“);* sFormatiert wird zu: „Wed Dec 18:30:00 2008“
System.IsVar	boolean system.IsVar(string name)	Prüft, ob eine Variable (Systemvariabe) definiert ist.	Legen Sie eine Systemvariable mit dem Namen vTEST an: *boolean bExist = system.IsVar(“vTEST“);* bExist ist true. *boolean bExist = system.IsVar(“vTESTEN“);* bExist ist false.
System.GetVar	var system.GetVar(string name)	Ermittelt den Wert einer Variablen.	Legen Sie eine Systemvariable mit dem Namen vTEST vom Typ Text an: *dom.GetObject(“vTEST“).State(“Testvariable“);* *string sTest = system.GetVar(“vTEST“);* sTest hat den Wert „Testvariable“. Wie in Teil 2 der Artikelserie beschrieben, können Sie den Wert einer Variablen auch folgendermaßen ermitteln: *dom.GetObject(“vTEST“).State(“Testvariable“);* *string sTest = dom.GetObject(“vTEST“).State();* sTest hat den Wert „Testvariable“.

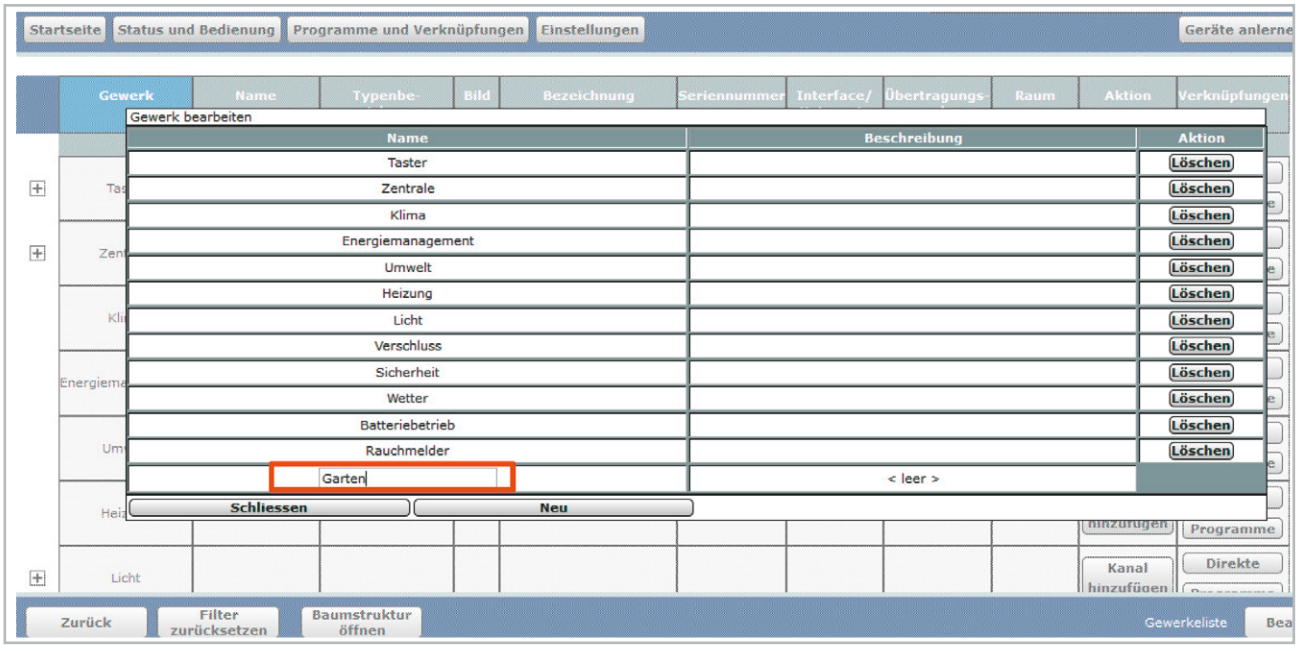
Scripte

Script „Alles im Garten ausschalten“

Mit den in Teil 1 der Artikelserie beschriebenen Befehlen zur Schleifenprogrammierung, den in Teil 2 besprochenen Objekten und der Möglichkeit, Aktoren (und/oder auch Sensoren) sogenannten Gewerken zuzuordnen, lässt sich eine Aufgabe, z. B. alles im Garten auszuschalten, sehr leicht in einem kleinen Script realisieren.

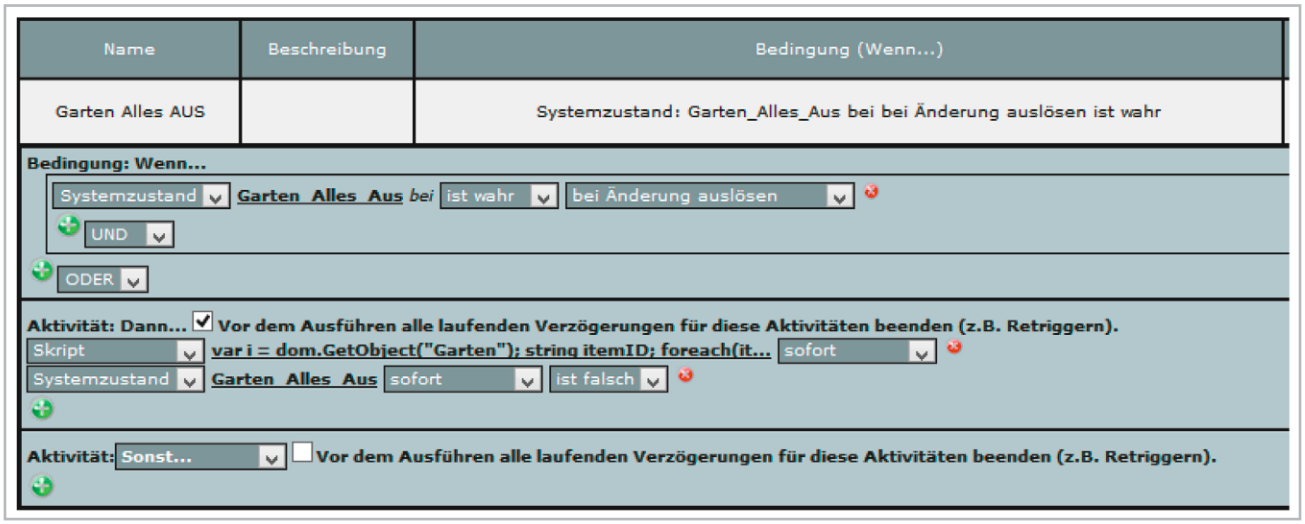
Das Script ist so aufgebaut, dass jederzeit Aktoren im Garten hinzugefügt oder weggenommen werden können, ohne dass das Script geändert werden muss.

Legen Sie zunächst über Einstellungen\Gewerke ein Gewerk mit dem Namen „Garten“ an:



... und ordnen Sie alle Elemente, die im Garten verbaut sind, diesem Gewerk zu.

Dann legen Sie ein Zentralenprogramm an, mit dem das Script später gestartet wird:



In diesem Zentralenprogramm wird das Script gestartet, sobald die (vorher anzulegende boolesche) Systemvariable *Garten_Alles_Aus* den booleschen Zustand *true* annimmt.

Nach dem Aufruf wird dann *Garten_Alles_Aus* wieder von unserem Zentralenprogramm auf *false* gesetzt.

Selbstverständlich kann man das Zentralenprogramm (und damit auch das Script) mit einem Taster oder einem beliebigen anderen Ereignis (z. B. Uhrzeit) auslösen. Eine Systemvariable zu nutzen bietet aber den Vorteil, ohne Änderung des Zentralenprogramms zum Ausschalten andere Zustände oder Ereignisse zum Auslösen später einbinden zu können.

Auch bei der Verwendung von Smartphone-Apps bietet diese Art der Programmierung Vorteile.

Das Script:

```
01  var i = dom.GetObject("Garten");
02  string itemID;
03  foreach(itemID, i.EnumUsedIDs())
04  {
05    var item = dom.GetObject(itemID);
06    if (item.IsTypeOf(OT_CHANNEL))
07    {
08      var device = dom.GetObject(item.Device());
09      if ((device.HssType().Find("HM-LC-Sw") >= 0) && (item.State()))
10      {
11        item.State(0);
12      }
13      if ((device.HssType().Find("HM-LC-Dim") >= 0) && (item.State()))
14      {
15        item.State(0.00);
16      }
17    }
18  }
```

In der Schleife zwischen Programmzeile 3 und 18 werden alle Aktoren vom Typ *HM-LC-Sw* (Unterputz-Schaltaktoren) und *HM-LC-Dim* (Dimmer), die dem Gewerk *Garten* zugeordnet sind, auf den Schaltzustand *true* (ein) geprüft (Zeile 9 und Zeile 13). Die Stelle *&& (item.State())* in Zeile 9 ist gleichbedeutend mit *&& (item.State()==true)*.

Sofern ein Aktor eingeschaltet ist, wird er ausgeschaltet (Zeile 11 und Zeile 15).

State-Machine (oder „endlicher Automat“)

Was ist eine State-Machine und wozu braucht man sie?

Eine State-Machine oder auch „endlicher Automat“ ist ein System, das zu jedem Zeitpunkt einen von endlich vielen stationären Zuständen (State) annehmen kann. Ereignisse (Events) sorgen dafür, dass die State-Machine von einem in den anderen Zustand versetzt wird. In den einzelnen Zuständen werden dann Aktionen ausgeführt.

Wozu dient nun eine State-Machine?

Typische Anwendungen sind z. B. die in der SPS-Technik angewandten sogenannten Schrittketten, die nichts anderes als State-Machines sind. Eine Gartenbewässerung beispielsweise, die Sektion nach Sektion im Garten wässern soll, wäre ein Beispiel für eine sinnvolle Anwendung in der Homematic Scriptprogrammierung.

Nehmen wir an, wir haben drei Sektionen im Garten („Oben", „Mitte", „Unten"), die nacheinander gewässert werden müssen, da der Wasserdruck nicht für die gleichzeitige Bewässerung aller drei Sektionen ausreicht.

Wir steuern die drei Sektionen jeweils über ein Ventil, das mit einem Homematic Aktor geschaltet wird. In der Hauptzuleitung zu den drei Ventilen liegt ein Hauptventil.

Für das Script müssen folgende Systemvariablen (Zentralenvariablen) vorher angelegt werden:

BewStatus Typ Zahlenwert – Bewässerungsstatus (der Zähler der Time-Machine)

Wert 0 → State-Machine steht, ist inaktiv (Initialisierungsschritt)
Wert 1 → Hauptventil EIN, Ventil Unten EIN, Bewässerung Unten läuft
Wert 2 und Ventil Unten ist ausgeschaltet → Ventil Mitte EIN, Bewässerung Mitte läuft
Wert 3 und Ventil Mitte ist ausgeschaltet → Ventil Oben EIN, Bewässerung Oben läuft
Wert 4 und Ventil Oben ist ausgeschaltet → Hauptventil AUS, Kette zurücksetzen mit BewStatus=0

BewGartenIstEin Typ boolean – Status der Gartenbewässerung für weitere Auswertungen

true = läuft
false = läuft nicht

BewZeitGarUnten Typ Zahlenwert – Bewässerungszeit für Sektion Unten in Minuten
BewZeitGarOben Typ Zahlenwert – Bewässerungszeit für Sektion Oben in Minuten
BewZeitGarMitte Typ Zahlenwert – Bewässerungszeit für Sektion Mitte in Minuten

Die drei Aktoren für die Sektionen heißen:

- Gartenbew:Unten
- Gartenbew:Mitte
- Gartenbew:Oben

Das Hauptventil heißt: Gartenbewaesserung

Das Script:

```
001 !                          DEKLARATIONSTEIL                          | Kommentar
002 !-------------------------------------------------------------------
003 !-------------------------------------------------------------------
004 !                    Status fuer State-Machine
005 !-------------------------------------------------------------------  | Hier wird der aktuelle Status (Zähler) in
006 integer Status = dom.GetObject("BewStatus").State();                  | die lokale Variable Status eingelesen ...
007 !-------------------------------------------------------------------
008 !Wenn Status = 0 (Startstatus), dann auf ersten Schritt setzen
009 !-------------------------------------------------------------------
010 if ( Status == 0 ) { dom.GetObject("BewStatus").State(1);}            | ... und, falls 0 auf den Wert 1 gesetzt.
011 !-------------------------------------------------------------------
012 !            Systemvariablen Bewaesserungszeiten
013 !-------------------------------------------------------------------  | Die in Systemvariablen gespeicherten Be-
014 integer nUnten = dom.GetObject("BewZeitGarUnten").State();            | wässerungszeiten für die 3 Sektionen wer-
015 integer nMitte = dom.GetObject("BewZeitGarMitte").State();            | den in lokale Scriptvariablen übertragen,
016 integer nOben = dom.GetObject("BewZeitGarOben").State();
017 !-------------------------------------------------------------------
018 !                          Ventile
019 !-------------------------------------------------------------------  | Es werden lokale Objekte aus den 3
020 object aktorUnten = dom.GetObject("Gartenbew:Unten");                 | Aktoren für die Bewässerung der 3
021 object aktorMitte = dom.GetObject("Gartenbew:Mitte");                 | Sektionen angelegt.
022 object aktorOben = dom.GetObject("Gartenbew:Oben");
023 !-------------------------------------------------------------------
024 !                    Hauptventil in Zuleitung
025 !-------------------------------------------------------------------  | Das Hauptventil ist den 3 Sektorventilen
026 object Hauptventil = dom.GetObject("Gartenbewaesserung");             | vorgeschaltet – wird ab hier über lokale
027 !-------------------------------------------------------------------  | Objektvariable angesprochen.
028 !                          PROGRAMM
029 !-------------------------------------------------------------------
030 !-------------------------------------------------------------------
031 !                       SEKTION 1 - UNTEN
032 !-------------------------------------------------------------------
033 ! wenn status = 1                                                     | Status = 1 → Sektion 1 wird bewässert.
034 !-----------------------------------
035 if ( Status == 1 ) {
036   !-----------------------------------
037   !Gartenbewaesserung laeuft setzen
038   !-----------------------------------
039   dom.GetObject("BewGartenIstEin").State(true);                       | → System- (Zentralen-) Variable setzen
040   !--------------------------
041   !Hauptventil EIN
042   !--------------------------
043   Hauptventil.DPByHssDP("STATE").State(true);                         | → Hauptventil einschalten
044   !---------------
045   !Zeit setzen                                                        | → Einschaltzeit für Sektionsventil 1 auf in
046   !---------------                                                    | Zentralenvariable gespeicherte Zeit
047   aktorUnten.DPByHssDP("ON_TIME").State(60.0 * nUnten);               | setzen. (*60, da Zeit in Minuten)
048   !---------------------
049   !Aktor einschalten
050   !---------------------
051   aktorUnten.DPByHssDP("STATE").State(true);                          | → Sektionsventil 1 einschalten
```

```
052   !-------------------
053   !neuen Status setzen                                    → System- (Zentralen-) Variable auf
054   !-------------------                                     neuen Status setzen. (Nächster Schritt,
055   Status = 2;                                             der dann mit Ausschalten des
056   dom.GetObject("BewStatus").State(Status);               Sektionsventiles 1 aktiv wird.
057   quit;                                                   → Script sofort verlassen
058 }
059 !-----------------------------------------------------------------
060 !                        SEKTION 2 - MITTE
061 !-----------------------------------------------------------------
062 !-----------------------------------------------------------------
063 ! wenn status = 2 und unten ausgeschaltet
064 !-----------------------------------------------------------------
65  if ( Status == 2 ) {                                      Status = 2 → Sektion 2 wird bewässert.
066   if ( aktorUnten.DPByHssDP("STATE").State() == false ) {
067     !---------------
068     !Zeit setzen                                          → Einschaltzeit für Sektionsventil 2 auf in
069     !---------------                                      Zentralenvariable gespeicherte Zeit
070     aktorMitte.DPByHssDP("ON_TIME").State(60.0 * nMitte); setzen. (*60, da Zeit in Minuten)
071     !----------------------
072     !Aktor einschalten
073     !----------------------
074     aktorMitte.DPByHssDP("STATE").State(true);            → Sektionsventil 2 einschalten
075     !--------------------
076     !neuen Status setzen                                  → System- (Zentralen-) Variable auf
077     !--------------------                                  neuen Status setzen. (Nächster Schritt,
078     Status = 3;                                           der dann mit Ausschalten des
079     dom.GetObject("BewStatus").State(Status);             Sektionsventiles 2 aktiv wird.
080   }
081   quit;                                                   → Script sofort verlassen
    }
    !-----------------------------------------------------------------
082 !                        SEKTION 3 - OBEN
083 !-----------------------------------------------------------------
084 ! wenn status = 3 und mitte ausgeschaltet
085 !-----------------------------------------------------------------
086 if ( Status == 3 ) {                                      Status = 3 → Sektion 3 wird bewässert.
087   if ( aktorMitte.DPByHssDP("STATE").State() == false ) {
088     !---------------
089     !Zeit setzen                                          → Einschaltzeit für Sektionsventil 3 auf in
090     !---------------                                      Zentralenvariable gespeicherte Zeit
091     aktorOben.DPByHssDP("ON_TIME").State(60.0 * nOben);   setzen. (*60, da Zeit in Minuten)
092     !----------------------
093     !Aktor einschalten
094     !----------------------
095     aktorOben.DPByHssDP("STATE").State(true);             → Sektionsventil 3 einschalten
096     !--------------------
097     !neuen Status setzen                                  → System- (Zentralen-) Variable auf
098     !--------------------                                  neuen Status setzen. (Nächster Schritt,
099     Status = 4;                                           der mit Ausschalten des Sektionsventiles
100     dom.GetObject("BewStatus").State(Status);             3 in diesem Fall die State-Machine
101   }                                                       zurücksetzt, da letzter Schritt)
102   quit;                                                   → Script sofort verlassen
103 }
104 !-----------------------------------------------------------------
105 !                              ENDE
106 !-----------------------------------------------------------------
107 ! wenn status = 4 und oben ausgeschaltet
108 !-----------------------------------------------------------------
109 if ( Status == 4 ) {                                      Status = 4 → Ende Bewässerung.
110   if (aktorOben.DPByHssDP("STATE").State() == false ) {
111     !--------------------------
112     !Hauptventil AUS
113     !--------------------------
114     Hauptventil.DPByHssDP("STATE").State(false);          → Hauptventil ausschalten.
115     !-------------------------------------
116     !Gartenbewaesserung laeuft zuruecksetzen
117     !-------------------------------------
118     dom.GetObject("BewGartenIstEin").State(false);        → System- (Zentralen-) Variable
119     !-------------------                                   rücksetzen.
120     !neuen Status setzen
121     !-------------------
122     Status = 0;                                           → Status State-Machine auf 0 setzen.
123     dom.GetObject("BewStatus").State(Status);
124   }
125   quit;                                                   → Script sofort verlassen (kann hier auch
126 }                                                         entfallen)
```

Die lokalen Variablen/Objekte (z. B. *aktorUnten, aktorOben, aktorMitte*) können natürlich auch weggelassen werden, wobei man dann anstelle

```
aktorUnten.DPByHssDP("STATE").State(true);
aktorUnten.DPByHssDP("ON_TIME").State(60.0 * nUnten);
```

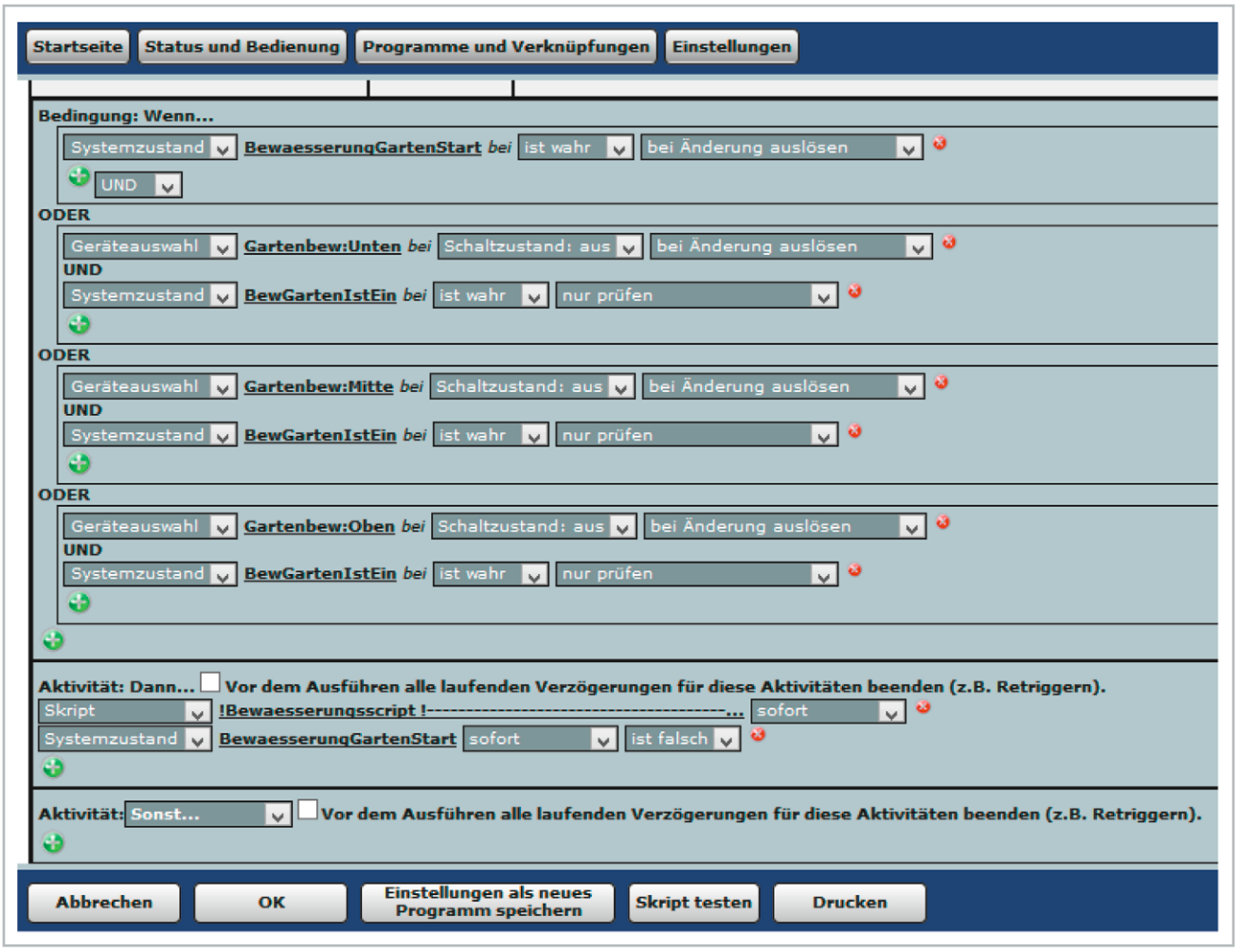
schreiben muss:

```
dom.GetObject("Gartenbew:Unten").DPByHssDP("STATE").State(true);
dom.GetObject("Gartenbew:Unten").DPByHssDP("ON_TIME").State(60.0 * nUnten);
```

etc. Das heißt, überall im Script wird `aktorUnten` durch `dom.GetObject("Gartenbew:Unten")` ersetzt.
Die Lesbarkeit wird allerdings damit schlechter.

Wann muss nun dieses Script aufgerufen werden? Da es für die Zustände *Oben ausgeschaltet, Mitte ausgeschaltet* und *Unten ausgeschaltet* jeweils keinen Status gibt, sondern im Script diese Zustände lediglich abgefragt werden, müssen die Aktionen „Ausschalten" der betreffenden Aktoren zum Starten des Scripts neben einer Systemvariablen *BewaesserungGartenStart*, die den Erststart der State-Machine durchführt, verwendet werden.

Das Zentralenprogramm für dieses Script sieht damit folgendermaßen aus:



Die Ausschalttrigger der drei Aktoren

werden mit der Systemvariablen (Zentralenvariablen) *BewGartenIstEin* verUNDet, damit bei einem manuellen Einschalten von einem der Aktoren die State-Machine nicht anläuft.

Ausblick

Im vierten Teil dieser Artikelreihe werden wir uns mit der Scriptprogrammierung, speziell im Zusammenhang mit dem CCU-AddOn CuxD, befassen.