



Schickes Smart Home

Visualisierung mit AIO CREATOR NEO und Smart Home Zentrale CCU3

Teil 10

In dieser letzten Folge der Reihe wird das Szenehandling mithilfe der Homematic CCU Scriptprogrammierung so erweitert, dass Szenen ohne Eingriff in die Programmierung verändert und gespeichert werden können.





Veränderbare Szenen

Die Szenensteuerung in der letzten Folge hat den Nachteil, dass sie nur durch Programmierung im mediola CREATOR verändert oder angepasst werden kann.

Wir wollen nun die Szenensteuerung mithilfe der Homematic Scriptprogrammierung so modifizieren, dass die einzelnen Szenen einfach verändert werden können, indem eine (Szenen-)Einstellung über die Lichtschalter oder Dimmer oder auch die Bedienungselemente auf der Visualisierung verändert/angepasst wird und durch eine Taste „Speichern“ abgespeichert und so – genau wie eingestellt – wieder abrufbar wird.

Die Umsetzung kann auch auf andere Art und Weise erfolgen, der Artikel soll einfach ein Beispiel dafür geben, was durch die Verbindung vom mediola CREATOR und der Homematic CCU in Verbindung mit einer Scriptprogrammierung möglich ist.

Benötigte Systemvariablen in der CCU

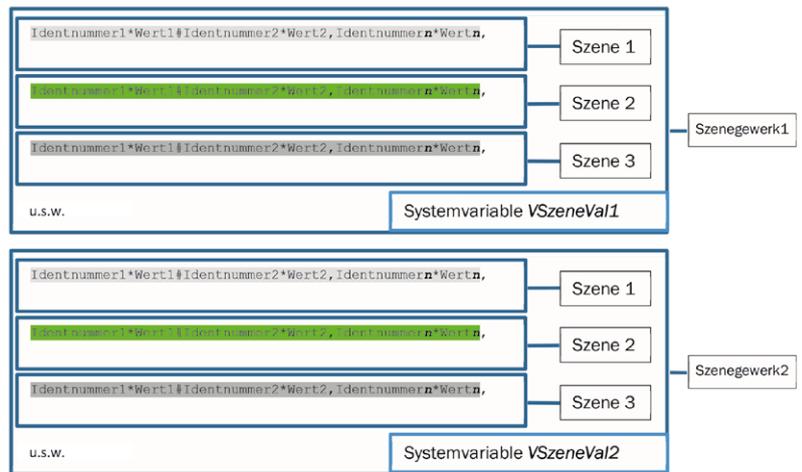
Da Werte von Schaltern (Ein/Aus) sowie Dimmern (Dimmwert in %) gespeichert werden müssen, benötigen wir zunächst „Speicherplätze“ für die Werte. Hierzu eignen sich natürlich die Systemvariablen (CCU Systemvariablen).

Wir können alle Werte aller Szenen eines Szenegewerks in jeweils eine einzige Systemvariable schreiben, wenn wir sie geschickt codieren. Für jedes Szenegewerk brauchen wir eine Systemvariable VSzeneVal mit dem Index des Szenegewerks, also VSzeneVal1 VSzeneValn (Bild 1).

Aufbau der Systemvariablen VSzeneVal zum Verständnis der Scripte:

Identnummer1*Wert1#Identnummer2*Wert2,Identnummern*Wertn,
 Identnummer1*Wert1#Identnummer2*Wert2,Identnummern*Wertn

Etwas übersichtlicher dargestellt:



usw.

Anmerkung: Die Größe der Daten sollte bei einem Wohnhaus ein Limit für die maximale Länge von Textvariablen nicht sprengen.

Das heißt, die Daten werden wie folgt in Gruppen zusammengefasst:

- Szenen sind durch Kommata getrennt
- Die Daten für einzelne Aktoren (Identnummer und Wert) einer Szene sind durch das Zeichen # getrennt
- Identnummer des Aktors und Wert des Aktors sind durch das Zeichen * getrennt
- Jedes Szenegewerk (1 ... n) besitzt eine Textvariable mit den Szenen für das Gewerk mit dem Namen VSzeneValn (n = Nummer des Szenegewerks, Bild 1)

Name	Beschreibung	Variablentyp	Werte	Maßeinheit	Kanalzuordnung
Filter		Filter		Filter	Filter
VSzeneVal1	Werte für Szenegewerk 1	Zeichenkette			
VSzeneVal2	Werte für Szenegewerk 2	Zeichenkette			
VSzeneVal3	Werte für Szenegewerk 3	Zeichenkette			
VSzeneVal4	Werte für Szenegewerk 4	Zeichenkette			
VSzeneVal5	Werte für Szenegewerk 5	Zeichenkette			

Bild 1: Systemvariable für Szenewerte



Startseite	Status und Bedienung	Programme und Verknüpfungen	Einstellungen							Geräte anlernen	Hilfe
Standby									hinzu- fügen	Programme	
SzeneGewerk1									Kanal hinzu- fügen	Direkte Programme	
SzeneGewerk2									Kanal hinzu- fügen	Direkte Programme	
Dimmer LED Sued Dimmaktor	HM-LC-Dim1PWM-CV		HM-LC-Dim1PWM-CV	LEQ0268744:1	Empfänger	Standard	Wohnzimmer	Löschen	Direkte Programme		
Dimmer LED WZEZ Dimmaktor	HM-LC-Dim1PWM-CV		HM-LC-Dim1PWM-CV	KEQ0558168:1	Empfänger	Standard	Wohnbereich	Löschen	Direkte Programme		
Dimmer Wohnzimmer:1 Dimmaktor	HM-LC-Dim1TPBU-FM		HM-LC-Dim1TPBU-FM	JEQ0205928:1	Empfänger	Standard	Esszimmer Wohnzimmer	Löschen	Direkte Programme		
LICHT_EINGANG_WZ_KUE Schaltaktor	HM-LC-Sw1-FM		HM-LC-Sw1-FM	LEQ0180382:1	Empfänger	Standard	Wohnzimmer	Löschen	Direkte Programme		
Stehlampe:1 Schaltaktor	HM-LC-Sw1-PI-DN-R1		HM-LC-Sw1-PI-DN-R1	MEQ0192174:1	Empfänger	Standard	Wohnzimmer	Löschen	Direkte Programme		
SzeneGewerk3								Kanal hinzu- fügen	Direkte Programme		

Bild 2: Gewerke (Szenegewerke)

Beispiel für Inhalt von VSzeneValn:

```
21250*0.000000#13745*0.700000#12081*false#12145*false#6814*false#14192*0.000000#34492*0.760000,21250*0.700000#13745*0.000000#12081*false#12145*false#6814*false#14192*0.660000#34492*0.000000,14192*0.990000#21250*0.980000#13745*0.980000#6814*false#12081*true#12145*false,,,,,34492*0.000000#21250*0.000000#13745*0.000000#12081*false#12145*false#6814*false#14192*0.000000
```

Die Szenenspeichervariablen werden als Stringvariablen (Zeichenketten) deklariert, da somit alphanumerische Zeichen und Zahlenwerte gespeichert werden können.

Wie schon erwähnt, benötigen wir eine Zuordnungsmöglichkeit für die Aktoren zu Szenen. Wir haben dies in der letzten Folge bereits angesprochen, die Gewerke nennen wir im Beispiel Szenegewerk1 ... n (Bild 2).

Mit diesen Szenegewerken ist es möglich, Aktoren (Schalter/Dimmer) quasi Bereichen in der Wohnung zuzuordnen, in denen Szenen dann wirken können. Diese Verfahrensweise hat den Vorteil, dass die Logik sehr flexibel wird. So ist es z. B. auch möglich, Szenen zu gestalten, die bereichsübergreifend sind (Bild 3).

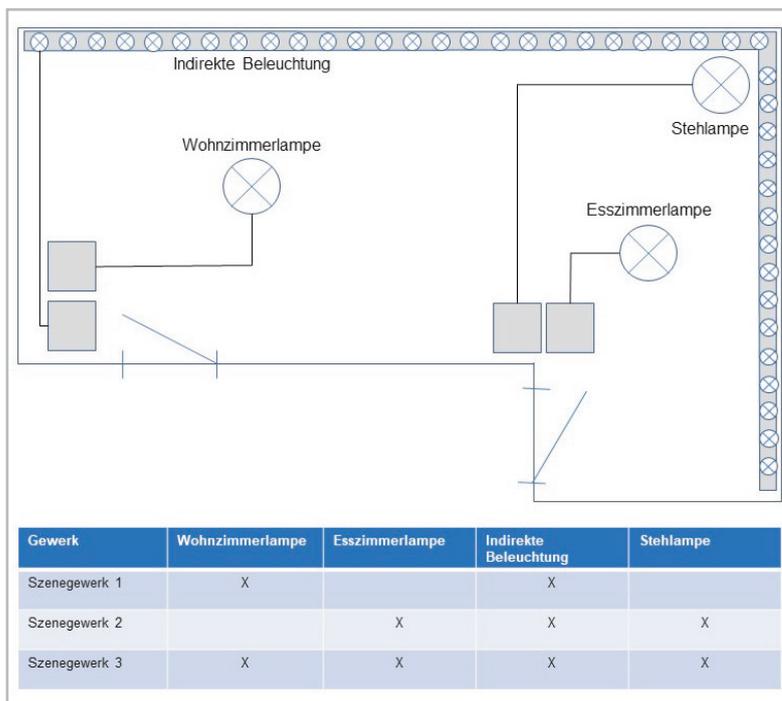


Bild 3: Beispiel Aktoren und Zuordnungen

Wir benötigen nun noch drei weitere CCU Systemvariablen sowie so viele VSzeneVal-Variablen, wie Szenegewerke verwendet werden:

Variable	Variablentyp	Funktion
SzeneSet	Zahl	Sobald diese Variable einen Wert > 0 annimmt, wird das Script zu Setzen einer Szene gestartet. Die Zahl kennzeichnet die Szenennummer.
SzeneOn	Zahl	Sobald diese Variable einen Wert > 0 annimmt, wird das Script zum Aktivieren (Einschalten) einer Szene gestartet. Die Zahl kennzeichnet die Szenennummer.
AktSzenegewerk	Zeichenkette	Diese Variable ist notwendig für die Anwendung der beiden oberen Variablen. Sie enthält den Namen des Szenegewerkes (Szenegewerk 1, Szenegewerk 2 ... Szenegewerk n), auf die sich die Szenennummern oben beziehen.
VSzeneVal1 ... n	Zeichenkette	Enthält die Typnummern und die Werte von Aktoren für alle Szenegewerke → wird von den Scripten genutzt. (n = Szenegewerksnummer).



Die Steuerseite im mediola CREATOR

Zunächst erweitern wir unsere Szenenseite um eine Tastatur zum Speichern der Szenen (wer die Seite weiter so nutzen möchte, fertigt eine Kopie der Seite an und arbeitet mit dieser).

Ob dies optisch genau so realisiert wird, ist eine Frage des persönlichen Geschmacks. Man kann sich auch vorstellen, dass es für die Anwahl von Szenen und die Programmierung/Speicherung von Szenen unterschiedliche Seiten gibt. Der Einfachheit halber gehen wir nun von der (kombinierten) Seite in [Bild 4](#) dafür aus.

Mit den neun Tasten im linken Bereich (überschrieben mit „SET Scene“) wird die Homematic Systemvariable SzeneSet mit einer Zahl 1–9 (entsprechend der gedrückten Taste) beschrieben.

Mit den 9 Tasten im rechten Bereich (überschrieben mit „Scene ON“) wird die Homematic Systemvariable SzeneOn mit einer Zahl 1–9 (entsprechend der gedrückten Taste) beschrieben.

Die Homematic Variable AktSzenegewerk wird beim Aufruf für das Szenegewerk 1 mit dem Text „Szenegewerk1“ beschrieben. Wenn mehrere unterschiedliche Szenegewerke verwendet werden sollen, werden entsprechend weitere Seiten erstellt, die entsprechend andere Aktoren beinhalten und bei deren Aufruf die Variable mit dem zugehörigen Szenennamen (Szenegewerk1, Szenegewerk2 Szenegewerkn) gesetzt wird.

Die Einrichtung der Seite ist im vorhergehenden Beitrag der Artikelreihe im ELVjournal 4/2020 [\[1\]](#) beschrieben worden.



Bild 4: Erweiterte Szenenseite

Die Scripte

Script 1 – Szene setzen

```

string itemID;
string sElement;
string sNeu;
integer n;
string sNewBlock;
string sOldBlock;
integer nAkt;

!-----
!SzeneschalterNr aus SysVar SzeneSet einlesen und in Zahl iSzSchNr (= Index der Szenedaten in !VSzeneVal) wandeln
!-----
string sSzSchNr = dom.GetObject("SzeneSet").State().Trunc().ToString();
sSzSchNr = sSzSchNr.Substr(0,sSzSchNr.Find("."));
integer iSzSchNr= sSzSchNr.ToInteger();

```



```
var myAssembly = dom.GetObject(dom.GetObject("AktSzeneGewerk").State());
string sBlock = "";

!-----
!Szenegewerknummer aus Systemvariable auslesen und zwischenspeichern
!-----
integer iStLaenge = dom.GetObject("AktSzeneGewerk").State().Length();
string sSzGewNr = dom.GetObject("AktSzeneGewerk").State().Substr(11,iStLaenge-11);
integer iSzGewNr= sSzGewNr.ToInteger();

!-----
!Gewerk Aktszenegewerk durchgehen und Werte des Eintrags iSzSchNr in sNewBlock merken
!-----
foreach(itemID, myAssembly.EnumUsedIDs())
{
    var item = dom.GetObject(itemID);
    if (item.IsTypeOf(OT_CHANNEL))
    {
        var device = dom.GetObject(item.Device());
        if ((device.HssType().Find("HM-LC-Sw") >= 0))
        {
            if (sBlock == "")
            {
                sBlock = itemID # "*" # item.State();
            }
            else
            {
                sBlock = sBlock # "#" # itemID # "*" # item.State();
            }
        }

        if ((device.HssType().Find("HM-LC-Dim") >= 0))
        {
            if (sBlock == "")
            {
                sBlock = itemID # "*" # item.State();
            }
            else
            {
                sBlock = sBlock # "#" # itemID # "*" # item.State();
            }
        }
    }
}

!-----
!Anzahl der aktuellen Bloecke in VSzeneVal feststellen
!-----
integer iAnzBlk = 0;
foreach (sElement, dom.GetObject("VSzeneVal" # sSzGewNr).State().Split(","))
{
    iAnzBlk = iAnzBlk + 1;
}

!-----
!Neuen String zum Speichern bilden
!-----
sNeu="";
! noch keine Eintraege
if (iAnzBlk == 0)
{
    if (iSzSchNr > 1)
    {
        !-----
        !Bis zur Nummer des Eintrag mit ,,," beschreiben
        !-----
        n=1;
        while (n<iSzSchNr )
        {
            sNeu = sNeu # ",";
            n = n +1;
        }
        sNeu = sNeu # sBlock;
    }
}
```



```

else
{
    sNeu = sBlock;
}
}
else
{
    ! schon Eintraege drin
    sNeu="";
    nAkt=1;
    sOldBlock="";

    foreach (sOldBlock, dom.GetObject("VSzeneVal" # sSzGewNr).State().Split(","))
    {
        if (nAkt<>iSzSchNr)
        {
            if (sNeu=="")
            {
                sNeu = sOldBlock;
            }
            else
            {
                sNeu = sNeu # "," # sOldBlock;
            }
        }
        else
        {
            If (nAkt==iSzSchNr)
            {
                if (sNeu=="")
                {
                    sNeu = sBlock;
                }
                else
                {
                    sNeu = sNeu # "," # sBlock;
                }
            }
        }
        nAkt = nAkt + 1;
    }
    if (iSzSchNr > iAnzBlk)
    {
        while(nAkt < iSzSchNr)
        {
            sNeu = sNeu # ",";
            nAkt = nAkt + 1;
        }
        sNeu = sNeu # sBlock;
    }
}

!-----
!Neue Werte schreiben
!-----
dom.GetObject("VSzeneVal" # sSzGewNr).State(sNeu);

```

Script 2 – Szene einschalten

```

!Szene aktivieren
!-----
!SzeneschalterNr aus SysVar SzeneON einlesen und in Zahl (= Index der Szenedaten in VSzeneVal) wandeln
!-----
string sSzSchNr = dom.GetObject("SzeneOn").State().Trunc().ToString();
sSzSchNr = sSzSchNr.Substr(0, sSzSchNr.Find("."));
integer iSzSchNr= sSzSchNr.ToInteger();

!-----
!Szenegewerksnummer aus Systemvariable auslesen und zwischenspeichern
!-----
integer iStLaenge = dom.GetObject("AktSzeneGewerk").State().Length();
string sSzGewNr = dom.GetObject("AktSzeneGewerk").State().Substr(11,iStLaenge-11);
integer iSzGewNr= sSzGewNr.ToInteger();

```



```

!-----
!Die entsprechende Aktorenlste aus dem File holen
!-----
string sListe = dom.GetObject("VSzeneVal" # sSzGewNr).State().StrValueByIndex("", iSzSchNr-1);

!-----
!Die einzelnen Elemente durchgehen
!-----
string part;
foreach(part, sListe.Split("#"))
{
    var itemIDStr = part.Substr(0,part.Find("*"));
    var item = dom.GetObject(itemIDStr);
    var valueStr = part.Substr(part.Find("*").ToInteger()+1;
    if ((valueStr != "false") && (valueStr != "true"))
    {
        item.State(valueStr);
    }
    else
    {
        if (valueStr == "false")
        {
            item.State(false);
        }
        if (valueStr == "true")
        {
            item.State(true);
        }
    }
}
}

```

Anmerkung: In den Scripten wird nicht abgefragt, ob Systemvariablen (Zentralenvariablen) angelegt wurden oder nicht. Ein Nichtanlegen von Systemvariablen im Vorfeld führt zu Abstürzen des Scripts.

In der Homematic CCU werden nun noch zwei Programme benötigt, eines zum Starten des Scripts für das Speichern von Szenen (Bild 5), ein zweites zum Starten (Aktivieren) von Szenen (Bild 6).

Mit der 9-er-Tastatur (rechter Bereich) auf der Szene-seite der mediola Visualisierung lassen sich die einzelnen Szenen einschalten. Alternativ oder zusätzlich können aber auch Homematic Komponenten (z. B. 6-fach-Schalter) verwendet werden, um die Szenen zu schalten. Dazu bedarf es lediglich noch eines Homematic Programms, welches in Bild 7 zu sehen ist.

Die weiteren Taster < 3 ... 6 > werden äquivalent programmiert. Dadurch, dass für jeden einzelnen Taster sowohl die Nummer der Szene als auch das Szenegewerk beim Drücken der Taste bestimmt werden, wird das Ganze äußerst flexibel. Es können 6-fach-Schalter (Bild 8) für ein Szenegewerk alleine oder übergreifend über mehr als ein Szenegewerk verwendet werden.

Mit diesem Beitrag beenden wir unsere Reihe zur Visualisierung mit mediola AIO CREATOR NEO und der Smart Home Zentrale CCU3. Unsere Beispiele zeigen nur einen kleinen Ausschnitt der Möglichkeiten, die mit der Kombination von Homematic Geräten und der mediola Oberfläche möglich sind.

The screenshot shows the HomeMatic Admin interface. At the top, there's a navigation bar with 'Admin' and 'Startseite > Programme und Verknüpfungen > Programme > Programmerrstellung'. Below that are tabs for 'Startseite', 'Status und Bedienung', 'Programme und Verknüpfungen', and 'Einstellungen'. The main content area shows a table with columns 'Name', 'Beschreibung', and 'Bedingung (Wenn...)'. A row is visible for 'Licht PSzeneSet' with the condition 'Systemzustand: SzeneSet im Wertebereich größer als 0.00 bei Änderung auslösen'. Below the table, there's a configuration section for the condition: 'Bedingung: Wenn...' with a dropdown for 'Systemzustand' set to 'SzeneSet im Wertebereich größer als 0.00' and an action dropdown set to 'bei Änderung auslösen'. There are also options for 'UND' and 'ODER' logic. The 'Aktivität: Dann...' section is checked, with a checkbox 'Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern)'. The script is set to 'string itemID; string sElement; string sNeu; integer n; stri...' and the action is 'sofort'. The system state is 'SzeneSet' with the action 'sofort' and a value of '0.00'. There's also a section for 'Aktivität: Sonst...' with a checkbox 'Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern)'.

Bild 5: Programm „Setzen von Szenen“



HomeMatic Admin
Startseite > Programme und Verknüpfungen > Programme > Programmerstellung

Startseite | Status und Bedienung | Programme und Verknüpfungen | Einstellungen

Name	Beschreibung	Bedingung (Wenn...)
Licht PSzeneOn		Systemzustand: SzeneOn im Wertebereich größer als 0.00 bei Änderung auslösen

Bedingung: Wenn...

Systemzustand **SzeneOn** im Wertebereich **größer als 0.00** bei Änderung auslösen

UND

ODER

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Skript **!Szene aktivieren!** dom.GetObject("I... sofort

Systemzustand **SzeneOn** sofort 0.00

Aktivität: Sonst... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Bild 6: Programm „Starten von Szenen“

HomeMatic Admin
Startseite > Programme und Verknüpfungen > Programme > Programmerstellung

Startseite | Status und Bedienung | Programme und Verknüpfungen | Einstellungen

Name	Beschreibung	Bedingung (Wenn...)	Aktivität (Dann...)
Szenenschalter		Kanalzustand: Wohnzimmer*6_fach_Taster*04:1 bei Tastendruck kurz	Systemzustand: AktSzeneGewerk

Bedingung: Wenn...

Geräteauswahl **Wohnzimmer*6_fach_Taster*04:1** bei **Tastendruck kurz**

UND

ODER

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Systemzustand **AktSzeneGewerk** sofort Szenegewerk1

Systemzustand **SzeneOn** sofort 1.00

Bedingung: Sonst, wenn...

Geräteauswahl **Wohnzimmer*6_fach_Taster*04:2** bei **Tastendruck kurz**

UND

ODER

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Systemzustand **AktSzeneGewerk** sofort Szenegewerk1

Systemzustand **SzeneOn** sofort 2.00

Bild 7: Szenen über Homematic 6-fach-Taster einschalten

Wir wünschen Ihnen viel Spaß beim Gestalten Ihrer eigenen, individuellen Bedienoberfläche und hoffen, Ihnen dafür einige Anregungen gegeben zu haben. **ELV**



Weitere Infos:

[1] ELVjournal 4/2020, Fachbeitrag Schickes Smart Home – Visualisierung mit AIO CREATOR NEO und Smart Home Zentrale CCU3, Teil 9: Bestell-Nr. 251432

- www.mediola.com
- Homematic IP Smart Home Zentrale CCU3 inkl. AIO-CREATOR-Lizenz und NEO Plugin AUTOMATION MANAGER: Bestell-Nr. 250407

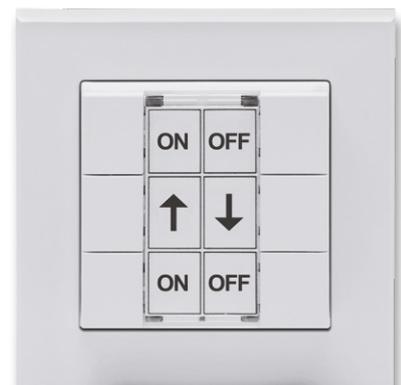


Bild 8: Homematic Wandtaster 6-fach