

# Skript für die Erfassung, Löschung, Ausführung von Befehlen

## von PC Oldie (Hans) V 1.00 vom 15.9.2017 16:30 Uhr

### 1. Leistungsumfang

Das Skript bezieht sich auf die Frage des Users Fonzo im Forum <https://www.symcon.de/forum/threads/35447-Timer-gesteuert-schalten>, ob es ein Skript gibt, welches folgende Aufgaben lösen kann:

- Schalte um 18.15 Uhr aus
- schalte in 35 Minuten aus
- Schalte um 20 Uhr für 2 Stunden an
- Schalte am nächsten Montag um 19 Uhr aus

Da ich mich mit dieser Fragestellung ebenfalls beschäftigt hatte, habe ich ein Skript entwickelt, welches für Anwender von IPSView in Verbindung mit HomeMatic Geräten eine Lösung für die genannten Fragestellungen aufzeigt. Auch Anwender mit Geräten anderer Hersteller können dieses Skript nutzen, indem sie die HM-Befehle durch die geeigneten Pendants ihrer Hersteller ersetzen. Da man bei IPSView letztendlich feste Adressen benötigt erscheint ein Skript, welches den Rahmen dafür bietet, die optimale Lösung zu sein, da man die Visualisierung ohnehin selbst vornehmen muss. Außerdem hat man damit die Freiheit das Skript so zu verändern, dass es den gewünschten Erfordernissen entspricht.

Von der Idee, Ereignisse zu erzeugen habe ich mich schnell verabschiedet, da die Implementierung zu viele Fragen und Probleme hinsichtlich der Reorganisation der erzeugten Ereignisse aufwirft. Stattdessen bin ich auf die Idee verfallen, die Verwaltung in Form einer Warteschlange zu realisieren wie es die Betriebssysteme selbst auch tun. Dies bedeutet, dass die auszuführenden Befehle in einem Array gespeichert und per Serialisierung in einem String in IPS gespeichert werden. Das Skript selbst wird jede Minute aufgerufen und prüft, ob ein Befehl ausgeführt werden muss. Dabei werden folgende Funktionen unterstützt:

- Das Schalten von Geräte Instanzen die über STATE oder LEVEL Datenpunkte verfügen, um Aktoren oder Dimmer ein- bzw. auszuschalten.
- Das Setzen von Booleschen Variablen auf TRUE bzw. FALSE.
- Die Ausführung von Skripten, womit sich letztendlich jede Art von Gerät steuern lässt.
- Die Deaktivierung und Aktivierung von Ereignissen.
- Die Steuerung der genannten Funktionen auch über Links.

Der Datumsbereich und die Anzahl der Befehle für die Ausführung können letztendlich durch die entsprechende Visualisierung selbst bestimmt werden. In den nachfolgend gezeigten Beispielen lässt sich das Datum um + 12 Wochen erhöhen und die Anzahl ist auf 9998 Befehle beschränkt. Im Beispiel können die Befehle 1-26 direkt gelöscht werden.

Um den Bedienungskomfort zu erhöhen und eine schnelle Eingabe der Daten zu ermöglichen, wurde die Möglichkeit der Befehlswiederholung geschaffen und die Eingabe eines Ein- und Ausschaltzeitpunktes mit einer Datensatzspeicherung.

### 2. Installation

Bevor man mit der Installation beginnt sollte das grundlegende Layout klar sein und eine Datensicherung erfolgt sein. Wichtig ist dabei die Anzahl der Tage die man in der View direkt zur Verfügung haben möchte. Mit der Installation wird die entsprechende Anzahl von Datumsfeldern für die Anzeige generiert. Will man später doch mehr oder wenige Tage haben, so kann man die Zahl im Skript nachträglich ändern. In diesem Fall werden die zusätzlich Variablen angelegt bzw. man kann überflüssige Variablen manuell löschen.

Zunächst ist im Objektbaum eine Kategorie anzulegen, die bei mir den Namen „Befehlsverwaltung“ trägt. Dort ein Skript erstellen – bei mir „Befehlsbearbeitung“ - und in dieses den Inhalt der Datei „Befehlsbearbeitung.php.txt“ mit einem kompatiblen Editor (Notepad++ o. ä.). kopieren und speichern.

Dann das Skript „Befehlsverwaltung“ direkt aus der Konsole heraus aufrufen. Damit wird das Skript „Befehlsbearbeitung Init“ generiert, welches in der Konsole seine ID ausgibt. In IPSView muss ein Image Button angelegt werden der diese ID erhält und damit auf „Befehlsverwaltung Init“ verweist. Ferner werden alle Variablen und das nicht aktivierte Ereignis für den minütlichen Abruf angelegt, wenn das Skript keine Unterobjekte hat. Das Ereignis muss dann später manuell aktiviert werden.

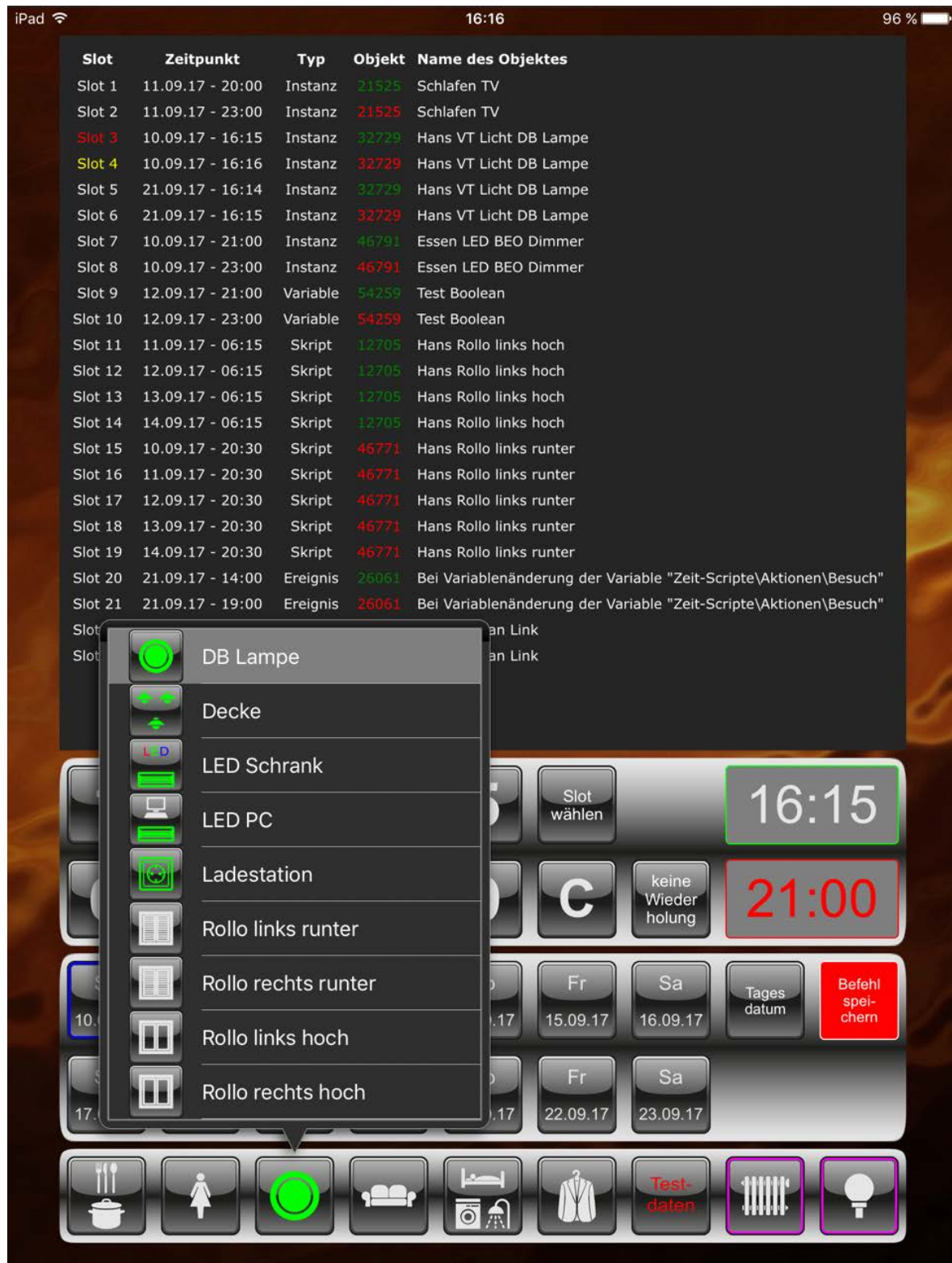
Unter Wochendaten findet man die Variablen Tag\_01 bis Tag\_07 für die Kürzel Mo–So und die Zeitstempel Datum\_01 bis Datum\_<\$AnzahlTage>.

27155		Befehlsverwaltung			
43547		Anzahl Wiederholungen	Integer	0	16:04:53
48427		Befehl Array	String	a:72:{i:0;s:22:"1,4,4239...	16:07:35
31360		Befehl Array HTML	String	<table style='margin:0 au...	16:14:56
57390		Befehl Array Speichern	Integer	0	16:14:56
29661		Index Tag	Integer	0	16:04:53
22584		Meldung	String		16:14:56
47423		Meldung Farbe	String	006400	16:02:59
22642		Meldung Verstecken	Boolean	True	16:14:56
19521		Objekt	Integer	0	16:04:53
57557		Shift Wochen	Integer	0	16:04:53
26293		Slot	Integer	13	16:12:39
47962		Slot Bis	Integer	20	16:14:56
43522		Slot Loeschen	Integer	0	Nie
42597		Slot Verstecken	Boolean	False	16:12:39
46481		Sortierung	Integer	2	16:04:37
21378		Zeiteingabe	Integer	0	16:04:53
12723		Zeitstempel An	Integer	0	16:04:53
14127		Zeitstempel Aus	Integer	0	16:04:53
43226		Zeitstring An	String	--:--	16:04:53
28244		Zeitstring An Farbe	String	00FF00	16:04:53
50902		Zeitstring Aus	String	--:--	16:04:53
28194		Zeitstring Aus Farbe	String		16:04:53
27977		Ziffer	Integer	4	16:02:57
53259		Ziffer Zaehler	Integer	0	16:04:53
27044		Wochendaten			
39246		Datum_01	Integer	1505426400	16:04:53
40259		Datum_02	Integer	1505512800	16:04:53
14401		Datum_03	Integer	1505599200	16:04:53
52357		Datum_04	Integer	1505685600	16:04:53
13019		Datum_05	Integer	1505772000	16:04:53
50571		Datum_06	Integer	1505858400	16:04:53
55997		Datum_07	Integer	1505944800	16:04:53
53166		Datum_08	Integer	1506031200	16:04:53
25051		Datum_09	Integer	1506117600	16:04:53
35200		Datum_10	Integer	1506204000	16:04:53
15912		Datum_11	Integer	1506290400	16:04:53
29442		Datum_12	Integer	1506376800	16:04:53
12631		Datum_13	Integer	1506463200	16:04:53
21991		Datum_14	Integer	1506549600	16:04:53
29370		Tag_01	String	Fr	16:04:53
23485		Tag_02	String	Sa	16:04:53
24457		Tag_03	String	So	16:04:53
57439		Tag_04	String	Mo	16:04:53
45143		Tag_05	String	Di	16:04:53
31536		Tag_06	String	Mi	16:04:53
10899		Tag_07	String	Do	16:04:53
39109		Befehlsbearbeitung	Skript	39109.ips.php	16:41:00
21705		Ereignis: Täglich alle 1 Minuten	Ereignis		16:41:00
55245		Befehlsbearbeitung Init	Skript	55245.ips.php	16:04:52

### 3. Aufbau der View und Zuordnung der Variablen

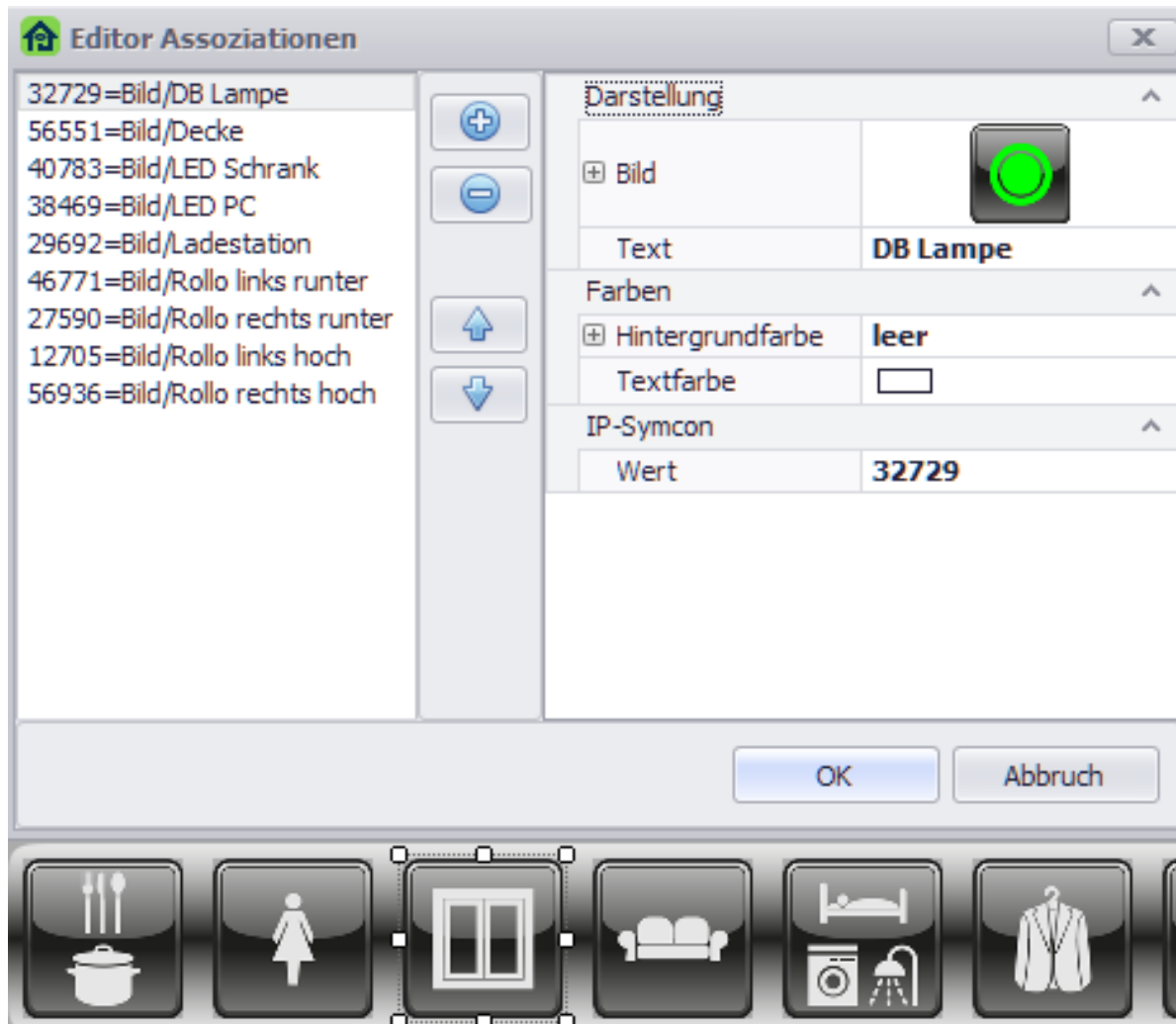
Das Bild zeigt die Gesamtübersicht der View auf einem iPad mit einigen Beispieldaten. Um die Oberfläche WAF-freundlich zu gestalten habe ich im unteren Bereich eine Auftrennung nach Räumen vorgenommen, so dass auch bei einer Vielzahl von Geräten die Übersicht nicht verlorengeht. Hier kann sich jedoch jeder konstruieren was immer er mag, da dies keinerlei Einfluss auf das Skript hat – es dient lediglich der Ordnung und der Sicherheit. Sicherheit deshalb, da ein Benutzer nur das auswählen kann, was vom Ersteller vorgegeben worden ist – Stichwort Tür plötzlich nachts offen!

Das Beispiel zeigt einige Geräte, die in meinem Zimmer verfügbar sind und die ich damit ansteuern kann. Die Rollos werden per Skript angesteuert und der Rest über HM Aktoren.



Technisch gesehen handelt es sich um Association Images, wobei der Text nur der Erläuterung dient, während das Image nach Auswahl in der View angezeigt wird. Die unter Wert eingetragene Nummer ist die ID des Skriptes, der Geräte Instanz, der Booleschen Variablen oder des Ereignisses, welches zum gewählten Zeitpunkt ausgeführt werden soll. Will man mehr mit Text arbeiten und diesen auch in der View anzeigen, dann sollte man statt Association Images auf Association Buttons ausweichen.

Die beiden Tasten mit den Rahmen in Magenta dienen bei mir lediglich zur Rückkehr auf andere Seiten.



Mit Aufruf des Skriptes werden die bereits abgearbeiteten Befehle automatisch entfernt. Im Übersichtsbild würde dies bedeuten, dass der Befehl auf Slot 3 in Rot nicht mehr angezeigt werden würde, da er bereits ausgeführt worden ist. Die Tatsache, dass er angezeigt wird beruht darauf, dass zum Zeitpunkt der Abarbeitung das Skript aktiv am Bildschirm lief.

Slot 4 ist in Gelb dargestellt was bedeutet, dass der Schaltvorgang nicht ausgeführt wurde. Möglicherweise war IPS deaktiviert oder das minütliche Abrufen des Skriptes zum Zeitpunkt der Ausführung war deaktiviert. Also eigentlich ein pathologischer Fall, da er nicht auftreten sollte und hier zu Testzwecken erzeugt wurde. Mit dem Start des Timers werden diese Befehle dann wieder ausgeführt. Will man dies verhindern, so sind diese vor dem Start des Timers manuell zu löschen.

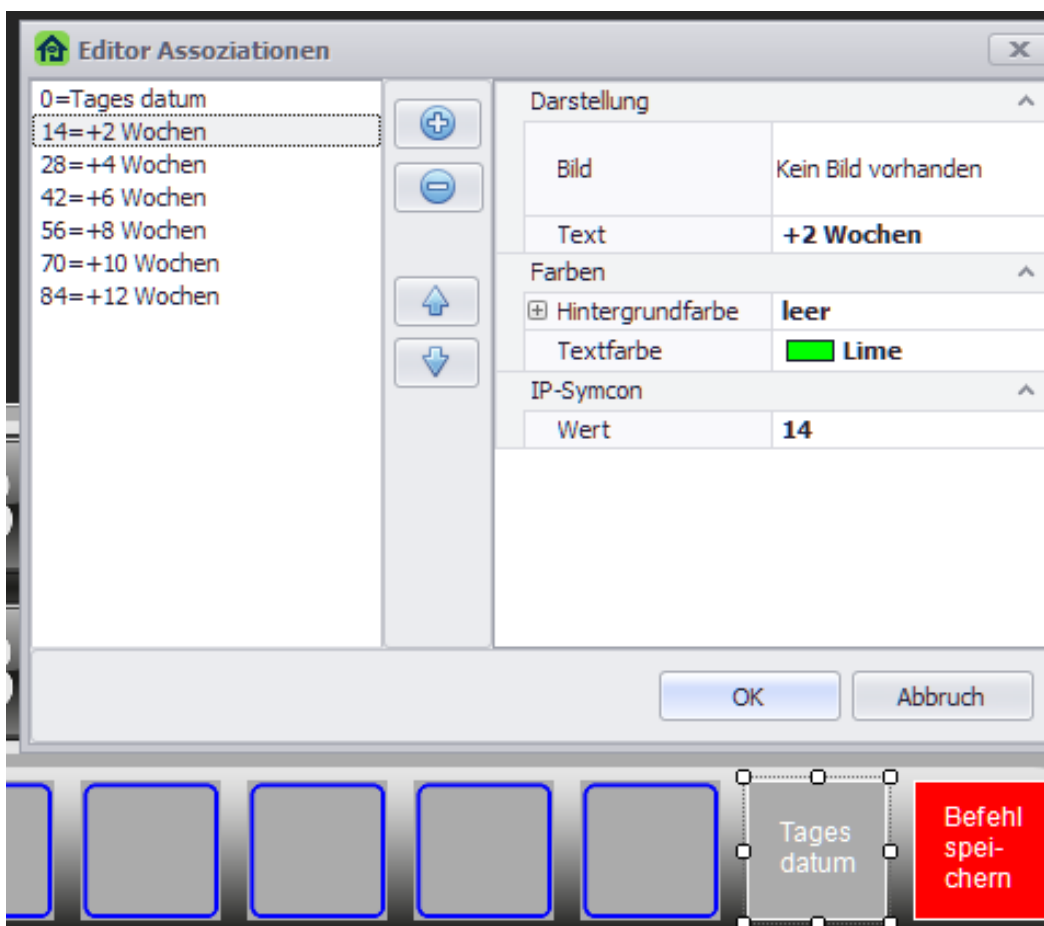
In meiner Darstellung werden 2 Wochen direkt für die Befehlseingabe zur Verfügung gestellt. Mit dem Aufruf des Skriptes erscheint das Tagesdatum links oben und ist mit einem blauen Rahmen versehen. Bei Bedarf kann der Zeitraum beliebig im Wochenrhythmus nach vorne verändert werden.





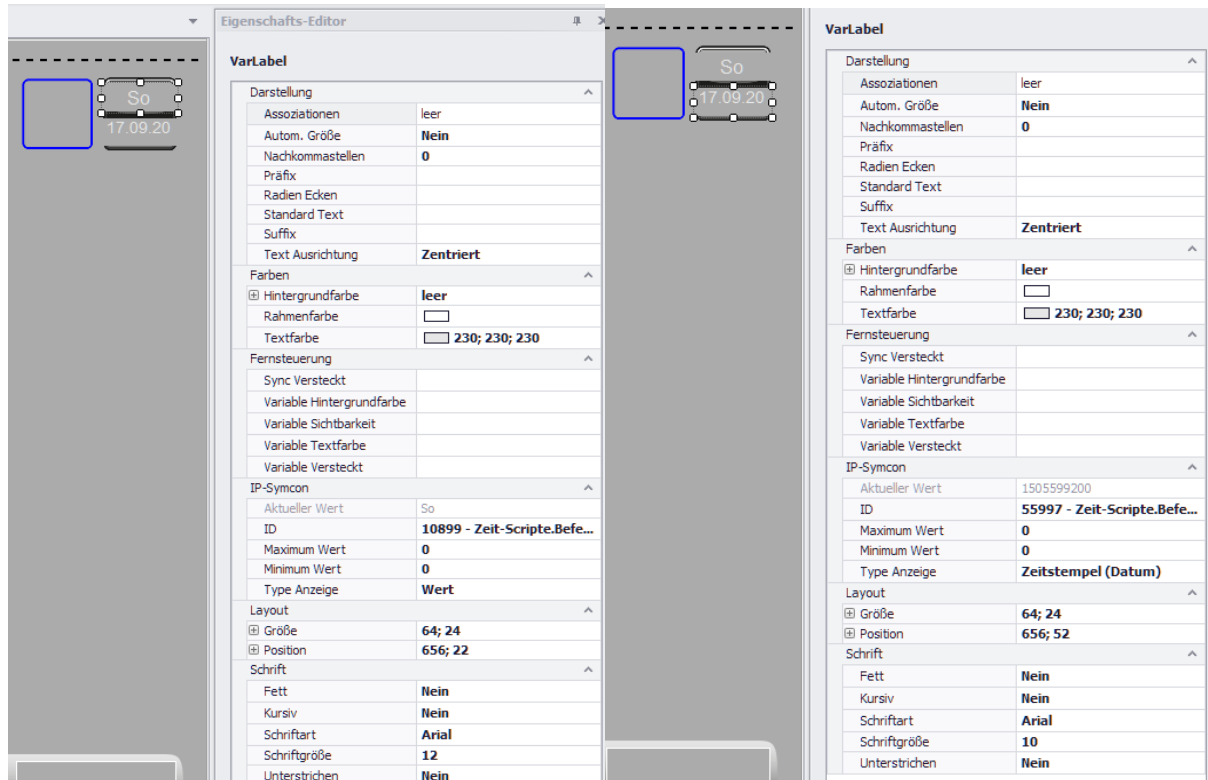
Ich habe mich für einen Zeitraum von + 12 Wochen entschlossen. Aber auch hier kann jeder hinterlegen was er möchte, ohne dass das Skript geändert werden muss. Entscheidend ist nur, dass die im Association Button hinterlegten Werte (= Tageszahl) wie im Beispiel mit + 2 Wochen = 14 Tage stimmen. Die Basis dafür muss immer 1 Woche (= 7 Tage) sein.

Auch ist es möglich statt der 2 Wochen nur 1 Woche oder auch z. B. über eine weitere Inline Seite ein ganzes Jahr anzeigen zu lassen. Dazu muss im Skript lediglich eine Variable angepasst werden, damit die entsprechenden Datumsfelder für die Anzeige in IPSView generiert werden → im Falle eines Jahres wären dies dann  $52 * 7 = 364$  Variablen.



**Auf der Seite 2 ist der Objektbaum dargestellt, so dass man erkennen kann, welche Variablen den einzelnen Feldern zugeordnet worden sind.**

Nachfolgend der Aufbau der Datumsfelder die in meinem Fall aus 3 Elementen bestehen. Im Hintergrund ist mein Standard Tastenbild. Darüber liegen die beiden Variablen Label die sich auf die generierten Tagesfelder Tag\_01 bis Tag\_07 und Datumsfelder Datum\_01 bis Datum\_<\$AnzahlTage> beziehen.



Das ausgewählte Bild wird durch ein Value Image dargestellt. Die zugewiesenen Werte an das Value Image sind bei den dargestellten 14 Tagen die Werte 0 – 13. Das jeweils ausgewählte Datum zeigt dann den darüber liegenden blauen transparenten Rahmen (Bild Ein) an, während die anderen Datumsfelder einen silbrigen transparenten Rahmen (Bild Aus) anzeigen.

