


Kurzanleitung Scriptdeveloper V2.xx.xx

Der Scriptdeveloper soll ein Hilfsmittel im Alltag bei der Erstellung von Homematic Scripten und deren Tests darstellen. Ein gewisses Wissen über Scripterstellung sowie den Aufbau einer CCU wird vorausgesetzt.

Die Software läuft auf Windows PC, ist bei nicht kommerzieller Nutzung Freeware und ist nicht an die Nichtverwendung einer Raspberrymatic oder sonstiger Einschränkungen gebunden.

Da mittlerweile aber schon einige tiefgreifende Operationen möglich sind, sind Mehrfachselektionen und Löschfunktionen erst nach Drücken von Unlock  zugänglich.

Trotzdem an der Stelle der Hinweis, welcher auch beim ersten Start des Programmes bestätigt werden muss:

Dies ist eine Testversion.

Die Verwendung dieser Software erfolgt auf eigenes Risiko
Der Autor dieser Software übernimmt keine Haftung für direkte oder indirekte Schäden, welche sich aus der Benutzung dieser Software ergeben sollten.
Eine kommerzielle Nutzung dieser Software ist untersagt

Ich bin einverstanden (Ja, Nein, wobei nein zum Programmende führt)

Hinweise über undokumentierte Methoden, die im Alltag nützlich sind aber ich bis jetzt auch noch nicht kannte, nehme ich gerne an und baue die auch gerne hier in das Programm mit ein.

Inhalt

| | |
|--|----|
| Kurzanleitung Scriptdeveloper V2.xx.xx | 1 |
| 1. Installation | 3 |
| 1.1 Systemvoraussetzungen | 4 |
| 2. Oberfläche | 5 |
| 3. Scripteditor | 6 |
| 4. Systemvariablen | 8 |
| 4.1 Sortieren | 10 |
| 4.2 Eigenschaften Systemvariablen | 10 |
| 4.3 Erweiterte Möglichkeiten | 11 |
| 4.3.1 Löschen von Systemvariablen | 13 |
| 4.3.2 Backup Systemvariablen | 13 |
| 4.3.3 Spezielle Filter | 16 |
| 5. Programme | 17 |

1. Installation

Das *.rar File in ein beliebiges Verzeichnis entpacken. Ein Installer ist nicht notwendig. Wichtig ist, in diesem Verzeichnis muss sich dann auch das Unterverzeichnis ico befinden (Die Grafiken für die Buttons).

In diesem Verzeichnis befindet sich auch das Konfig File. Bei der erstmaligen Verwendung muss dieses angepasst werden

```
[VALUES]
CCUIP=192.168.x.x      ← IP der CCU
SSHPW=SSHPASSWORD    ← Das verwendete SSH Passwort des CCU
PSCP=pscp
ALLOW=Black Testversion
[LAST]
THIS=$this$
SCR=$scr$
DATEI=C:\MTH\Homematic\sysvars.hsc
```

Der Scriptdeveloper nutzt pscp, um auf die Logdatei der CCU zugreifen zu können. Aus diesem Grunde muss für das Programm auf der CCU / Raspberrymatik SSH freigegeben sein und das ssh Kennwort muss bekannt sein, intern wird via pscp die Logdatei geholt zum Auswerten.

pscp (gehört zur putty Suite) muss also auf dem PC vorhanden und installiert sein

pscp kann allerdings bei der Erstmöglichen Verwendung einen Fallstrick bergen. Genau wie putty beim erstmaligen Verwenden eine Bestätigung abfragt, dass das Serverzertifikat der CCU unbekannt ist und ob diesem vertraut werden soll so stellt auch pscp diese Abfrage. bei einem Start als hidden innerhalb meines Programmes kann der Anwender diese Abfrage allerdings nicht machen und so hängt dann der Script-Developer.

Aus diesem Grunde vor der erstmaligen Verwendung (oder denselben Wechselgründen wie bei der CCU unter putty auch) unter CMD einmal die folgende Kommandozeile aufrufen: (Dies ist die gleiche Codesequenz die auch der Script Developer aufruft)

```
pscp -pw DeinSSHPASSWORT root@192.168.X.X:/var/log/messages vorher.txt
192.168.X.X= IP deiner CCU
```

Die Abfrage in der Kommandozeile, ob dem Key des Servers vertraut werden soll, einmalig mit y bestätigen, danach funktioniert der Zugriff.

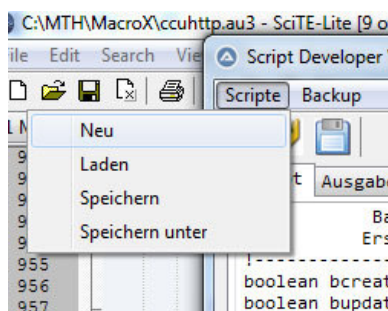
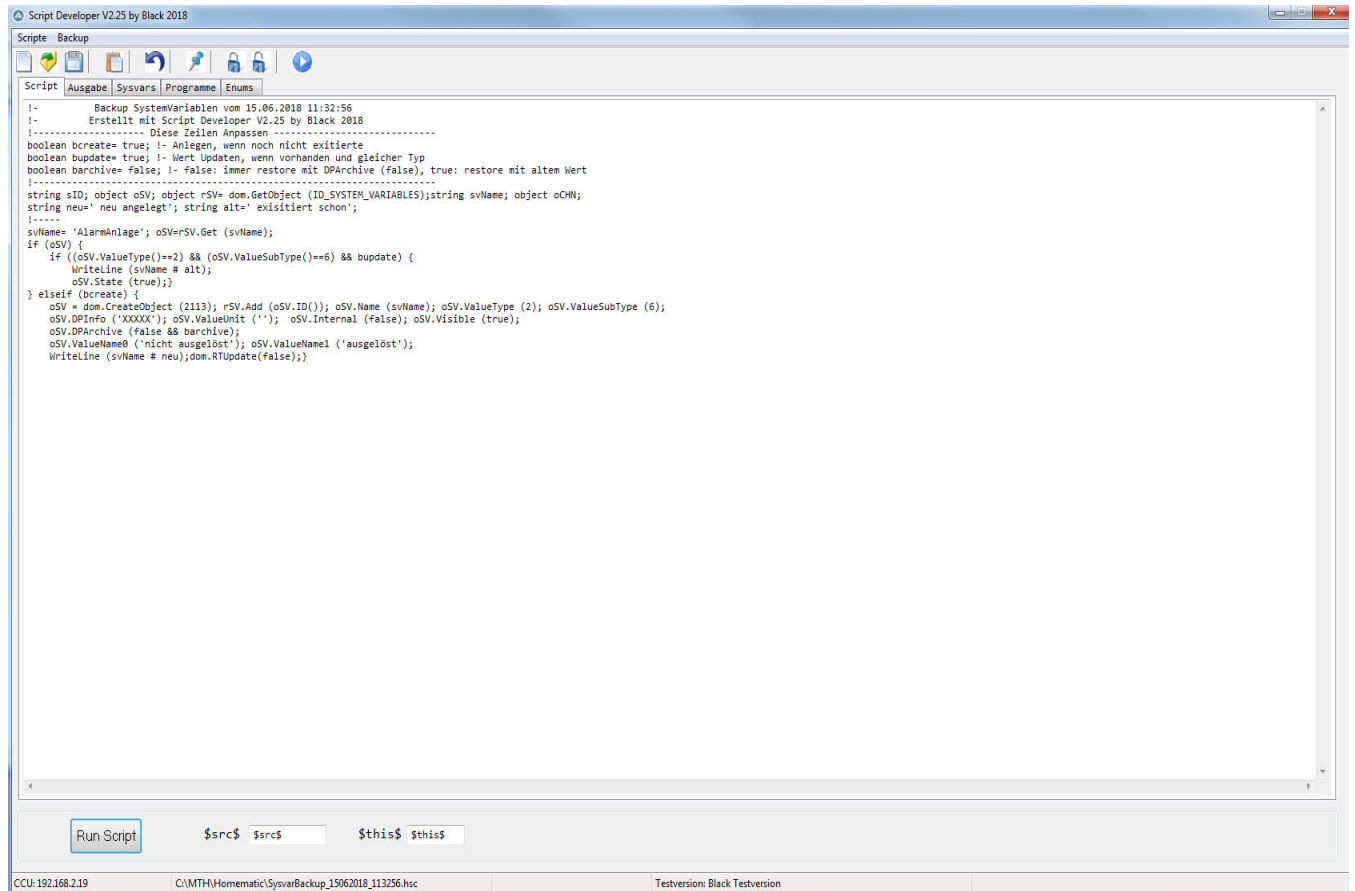
1.1 Systemvoraussetzungen




Der Scriptdeveloper lief bisher in Testinstallationen unter WIN 7 64/32 bit, Win 8.1 64 bit und unter Win 10 64bit. Da unter recht konservativen Compilereinstellungen übersetzt wurde, sollte er eigentlich unter allen Windows Version laufen (ab Win 7)

Auf der Homematic-Seite wurde bei mir auf einer Raspberrymatic 2.31.25.20180428 getestet.

Auf einer CCU sind die erzeugten internen Progs auch lauffähig, wenn Rega-Community eingestellt wird. Unter Legacy läuft es NICHT !

2. Oberfläche




Im Menüreiter Scripte finden sich die Einstellungen zum Anlegen eines neuen Scriptes , zum Laden eines bestehenden Scriptes  und zum Speichern eines Scriptes im Scripteditor  sowie speichern unter.

In der Statuszeile finden sich Information über:

1. IP der Host CCU
2. Dateinames des Scriptes im Scripteditor
3. Result des letzten Aufrufes (normalerweise 200, OK)
4. Versionshinweis
5. Clipboard Inhalt (dazu später mehr)

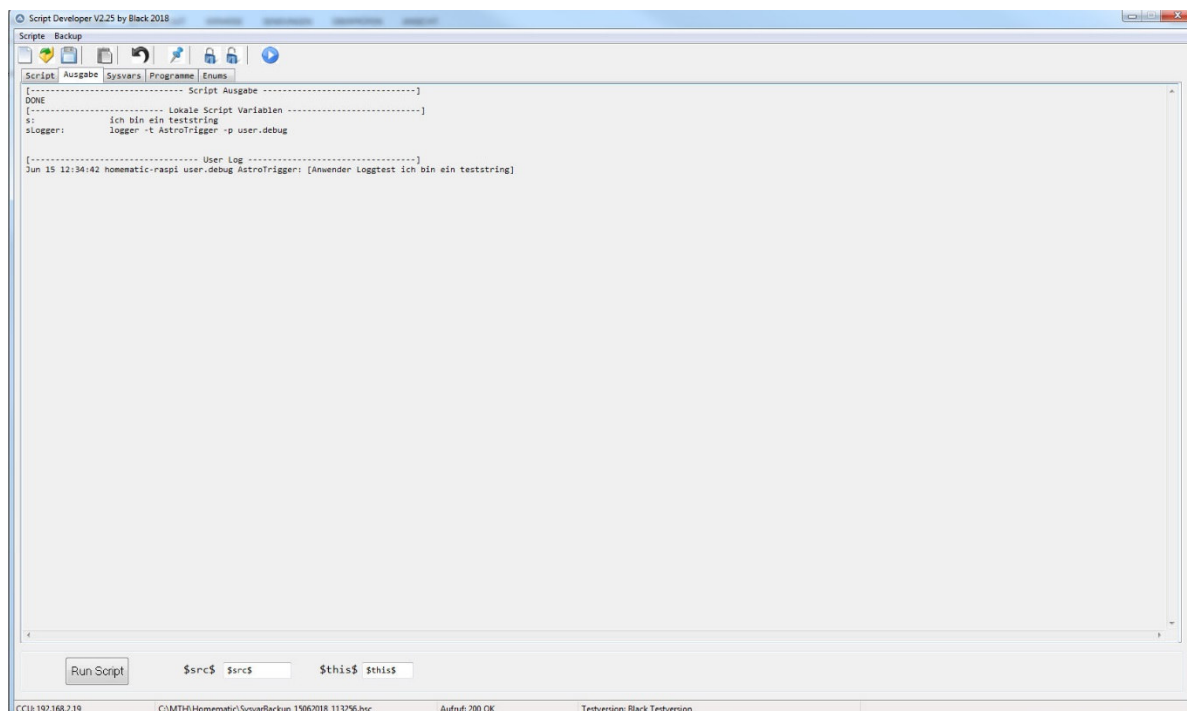
3. Scripteditor

Im Scripteditor werden die Scripte geschrieben oder geladen, die mittels Run Script oder  an die CCU zum Ausführen gesendet werden. Das Scriptergebnis wird dann im Reiter Ausgabe angezeigt. Dieses kleine TestScript zum Beispiel:

```
string s= "ich bin ein teststring";
string sLogger      = "logger -t AstroTrigger -p user.debug ";

datapoints.Get("CUXD.CUX2801001:1.CMD_EXEC").State (sLogger # "[Anwender
Loggtest " # s # "]");
WriteLine ("DONE");
Hier testweile aus State
```

Erzeugt folgende Ausgabe:



Script Ausgabe stellt alles dar, was in dem Script mit Write, WriteLine oder Derivaten zur Ausgabe gebracht wurde,

Unter lokale Script variablen stehen die Variablen welche im Script definiert wurden mit ihren Namen. In dem Fall hier sind das die Beiden String Variablen s und sLogger.

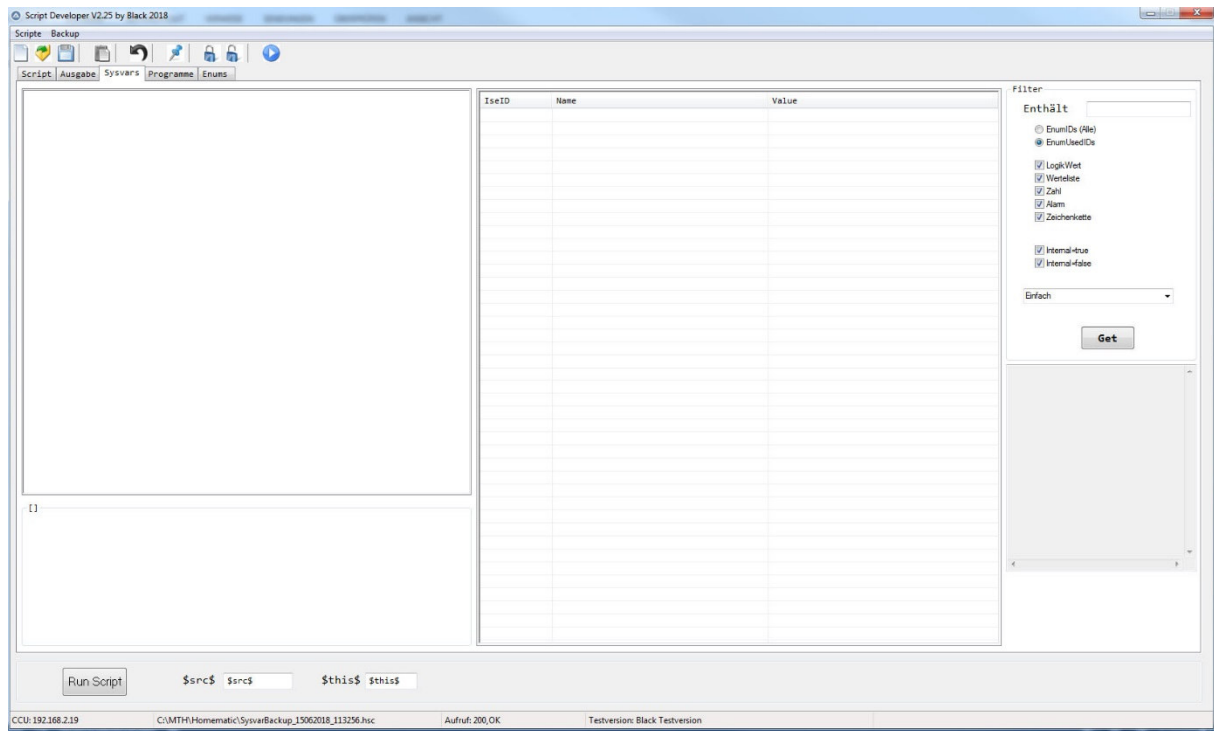
Wurde via Userlog ein Eintrag im Logfile erzeugt, so wird dieser nach Scriptende auch hier angezeigt. Dafür braucht es u.a. den pscp.

Sollte in dem Script ein Fehler sein (hier testweise State zu Stat geändert) erhält man die gleiche Ausgabe wie im Syslog:

```
[----- Fehler im Script -----]
Jun 15 12:41:49 homematic-raspi local0.err ReGaHss: Error: IseESP::SyntaxError= Error 1 at
row 4 col 88 near ^ (sLogger # "[Anwender Loggtest " # s # "]);^M WriteLine ("DONE");^M
[iseESP.cpp:1121]
Jun 15 12:41:49 homematic-raspi local0.err ReGaHss: Error: ParseProgram: SyntaxError=
(sLogger # "[Anwender Loggtest " # s # "]);^M WriteLine ("DONE"); [iseESP.cpp:374]
```

4. Systemvariablen

Im Reiter Systemvariablen können Systemvariablen gefiltert angezeigt werden.



Filteroptionen:

Enthält: leerer Eintrag = es wird nicht nach enthaltener Buchstabensequenz selektiert
Eingegebener Text. Die Systemvariable muss im Namen die Buchstabensequenz enthalten.

EnumIDs : es werden alle Systemvariablen durchsucht

EnumUsedIDs: es werden nur die Systemvariablen durchsucht, welche auch in der WebUI dargestellt werden

Anwahlmöglichkeit:

Es werden nur Systemvariablen dargestellt welche:

Logikwert (boolean)

Werteliste (integer)

Zahl (float)

Alarm (Alarmvariable)

Zeichenkette (string) sind.

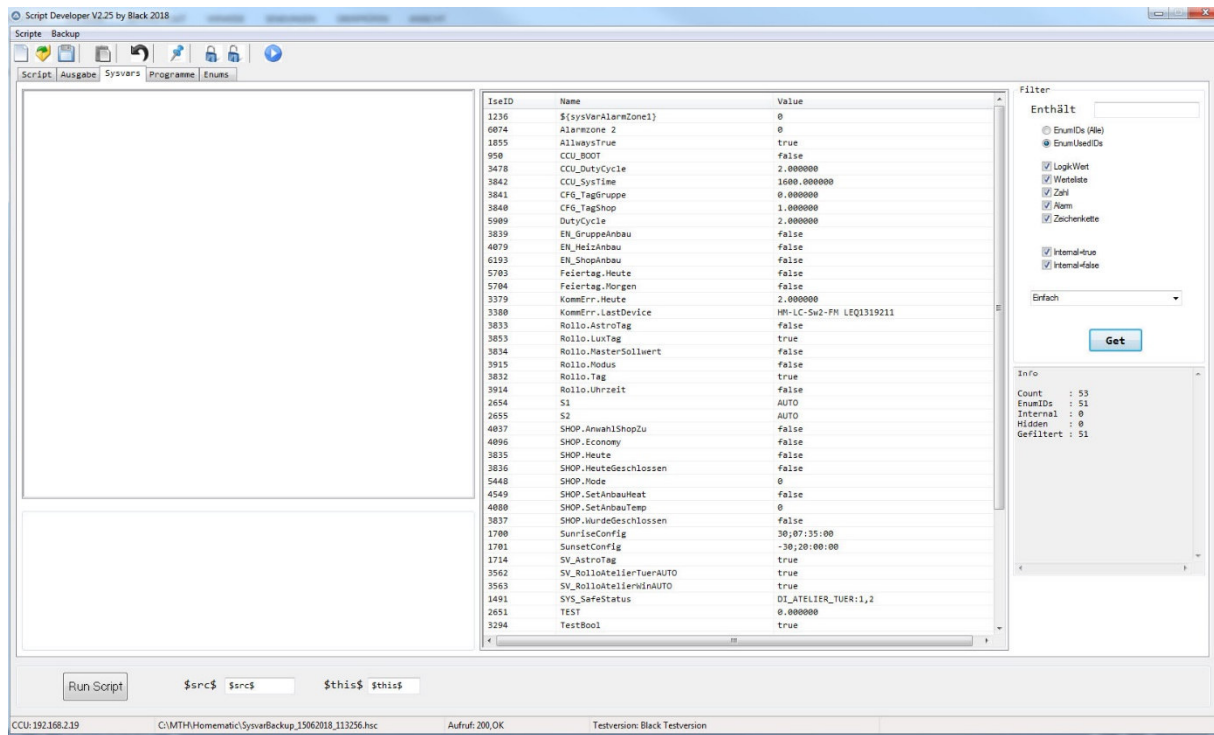
Mehrfachselektion ist hier möglich

Mit Internal=true werden gezielt nur als Intern markierte Systemvariablen dargestellt

Mit Internal=false werden die Systemvariablen dargestellt, welche nicht intern sind.

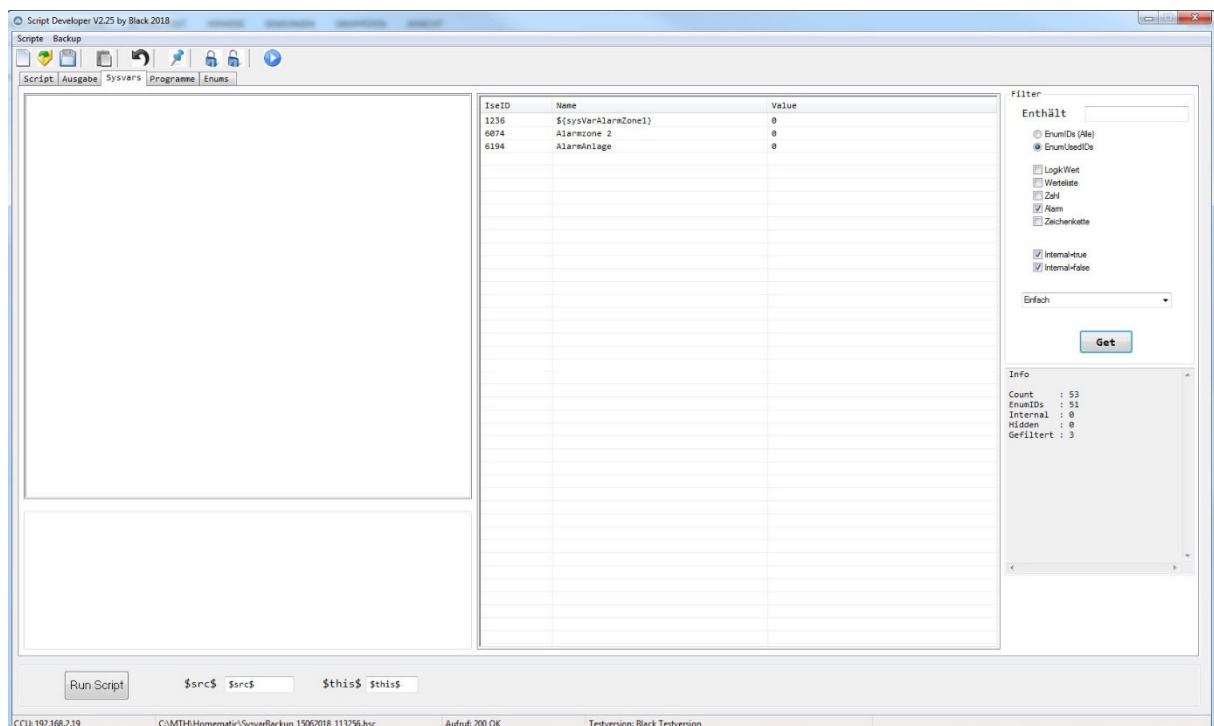
Zusatzselektion ist möglich über das Auswahlfeld, welches bis jetzt die einstellung einfach und „verwaiste Channel“ ermöglicht. Einfach ist die Standarteinstellung.

Durch Druck auf Get wird die Liste gemäß Selektion von der CCU angefordert, aufbereitet und dargestellt



Im Infocfeld verraten schon die Zahlen Count = 53 und EnumIDs (gewählte Filterart) =51, das sich dort noch 2 Systemvariablen verstecken. Diese Darstellung entspricht der Darstellung der WebUI.

Suche ich zum Beispiel nur nach Alarm variablen: wäre dies die gewünschte Einstellung:

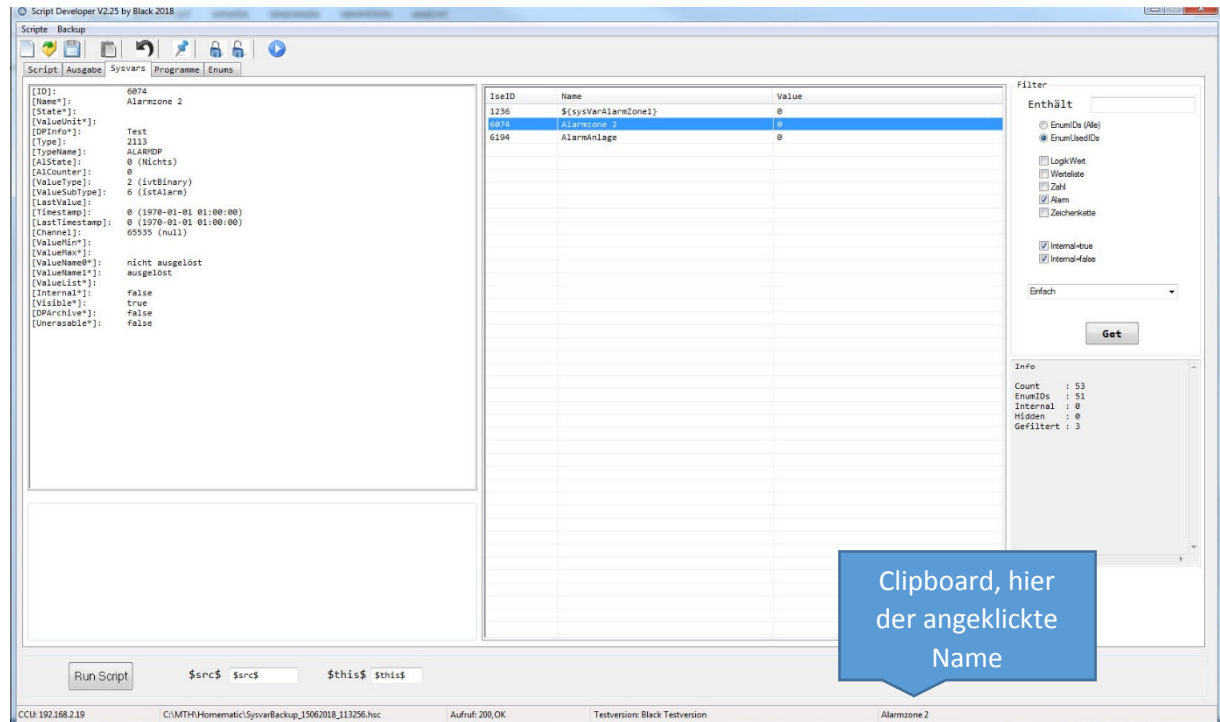



4.1 Sortieren

Durch Drücken auf ISEID oder Name kann das Ergebnisfeld nach ID oder Namen Sortiert werden.

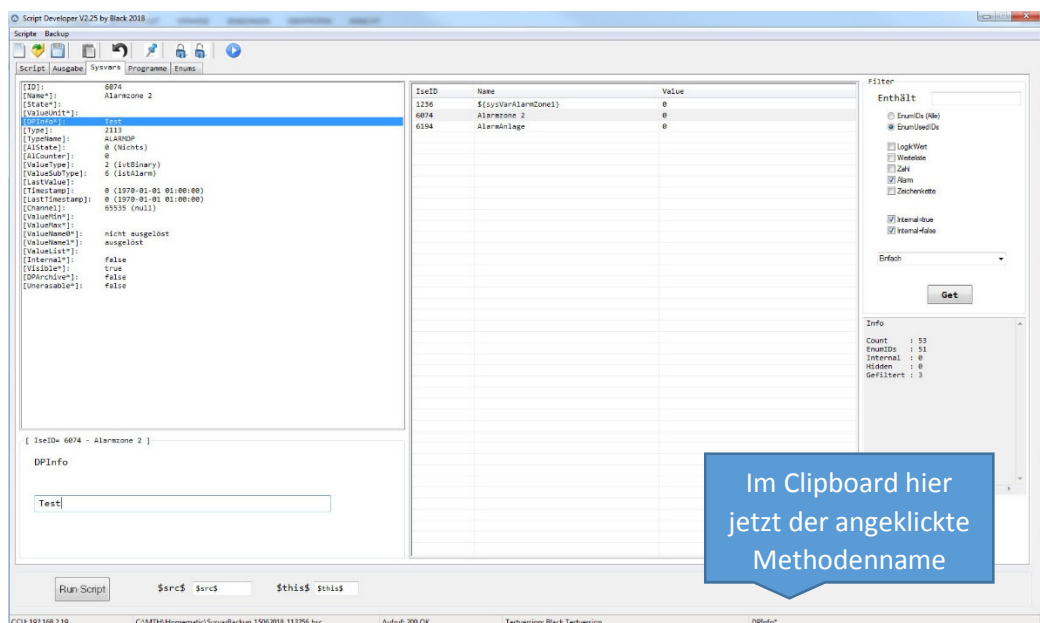
4.2 Eigenschaften Systemvariablen

Durch Doppelklick auf die Systemvariable wird das Eigenschaftsfeld mit Werten befüllt



Im Scripteditor lässt sich durch Drücken von  der Inhalt des Clipboard an die aktuelle Cursorposition einfügen (Vermeidung von Schreibfehlern)

Im Eigenschaftsfenster lassen sich alle mit einem * gekennzeichneten Eigenschaften im Scriptdeveloper Editieren



Warnung: eine Änderung von Werten in diesem Fenster führt zu einer sofortigen Änderung von Werten auf der CCU. !!!

Sollte sich die Eingabe nicht Abschließen lassen (Zeichen anstatt Zahlen eingegeben bei Float oder Integer) bleibt das Fenster weiterhin offen.

4.3 Erweiterte Möglichkeiten

Die hier nun beschriebenen Möglichkeiten sind erweitert und müssen explizit Freigeschaltet werden

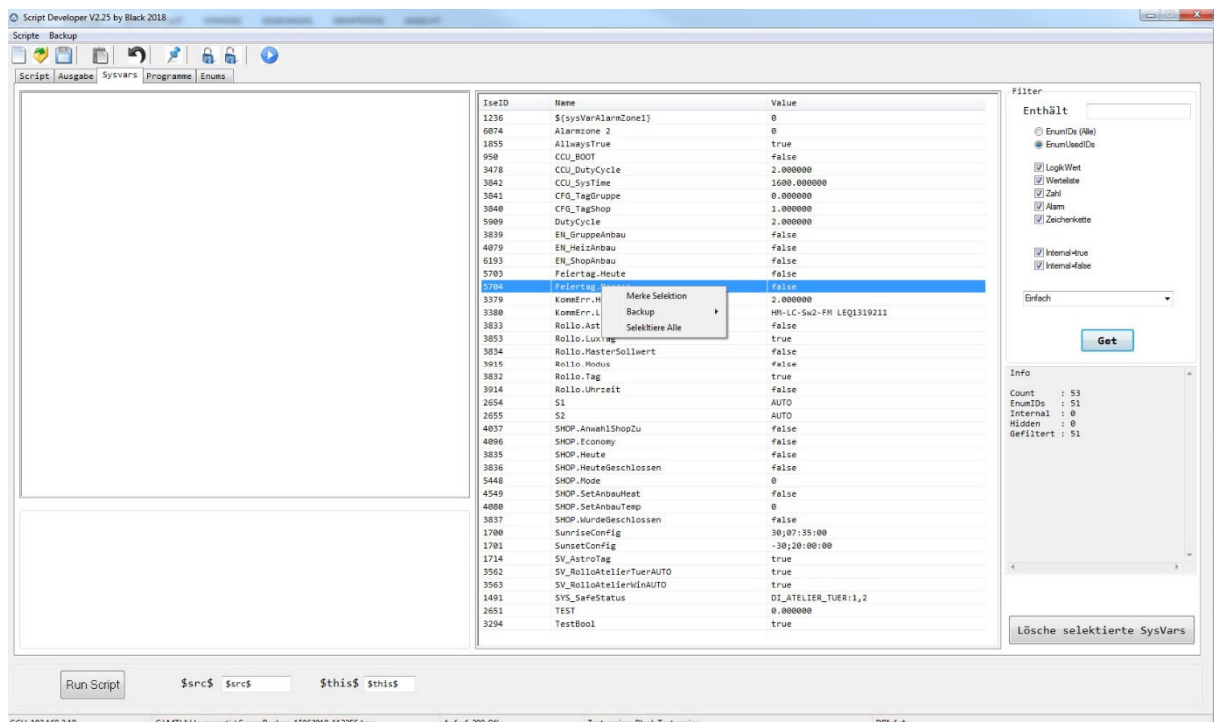



Schloss zu: Einfache Bearbeitung anwählen

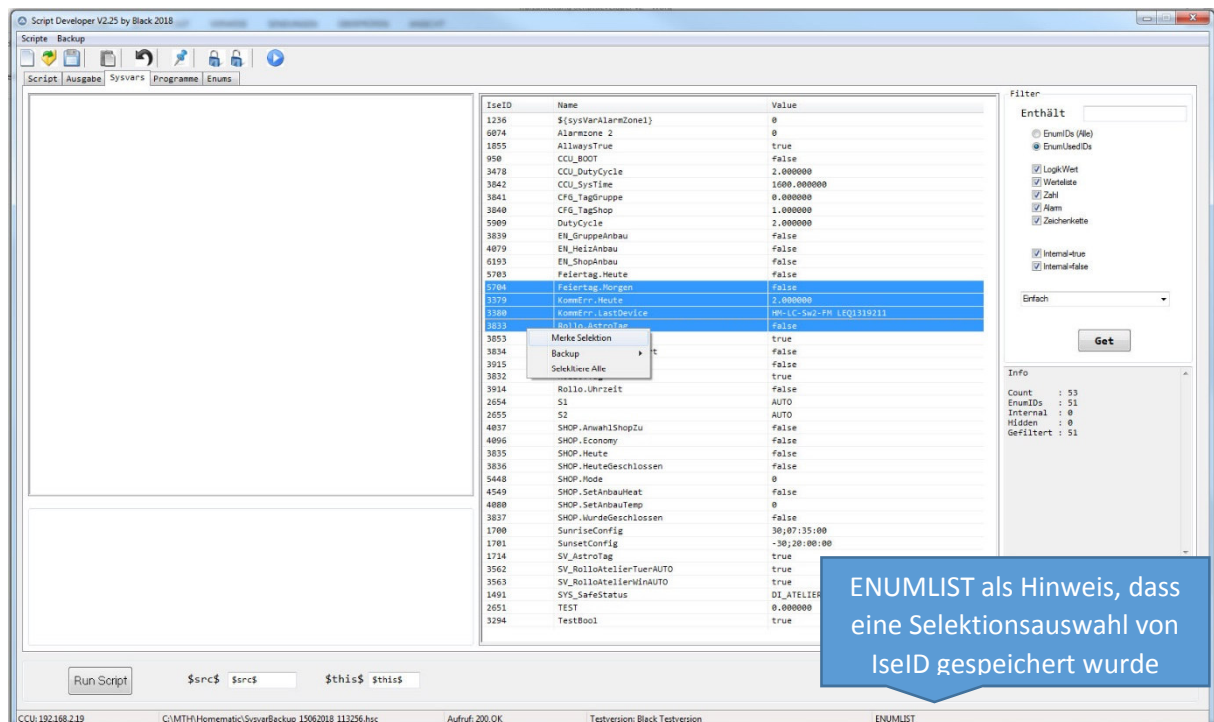


Schloss auf: Erweiterte Bearbeitung anwählen


In dem ListFeld der Systemvariablen ergibt sich bei Rechtsklick der Maus folgendes Menü:



Die IselID Liste der Selektierten Systemvariablen wird gemerkt. Alternativ dazu gibt's auch den  Button. Erweitert ist auch Mehrfachsektion möglich.



Diese Selektionsauswahl kann benutzt werden, wenn man mit den bestimmten, ausgewählten Systemvariablen Aktionen durchführen möchte.

Geht man jetzt in den Scriptedit, neues Script und drückt  so wird gemäß der gerade gemachten Auswahl folgendes kleines Programmchen erzeugt:

```
string sID;
object oID;
string sEnum="5704\t3379\t3380\t3833";
foreach (sID,sEnum) {
    oID=dom.GetObject (sID);
}
```

Innerhalb der Schleife müssen nun nur noch die Anweisungen programmiert werden. Als kleines Helferlein im Alltag..

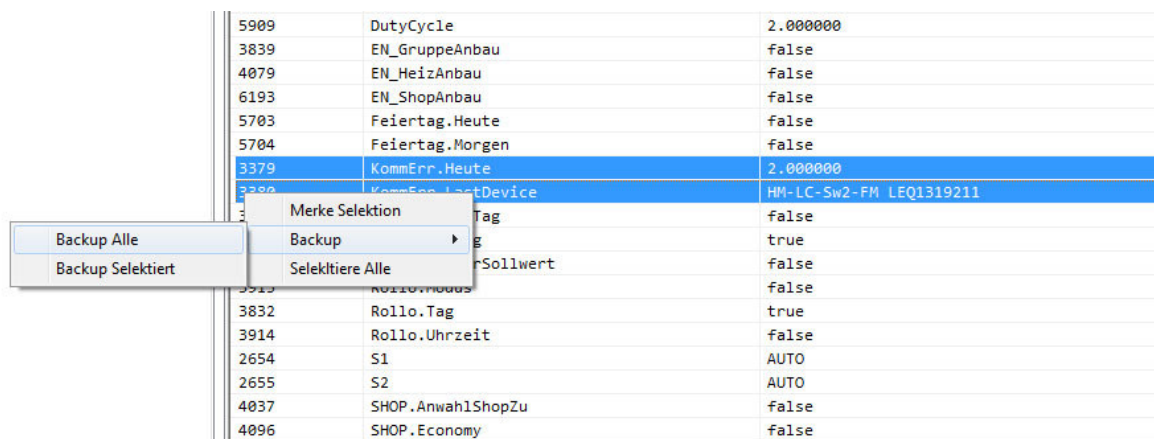
Selektiere Alle: selektiert alle im Auswahlfenster angezeigten Systemvariablen.

4.3.1 Löschen von Systemvariablen

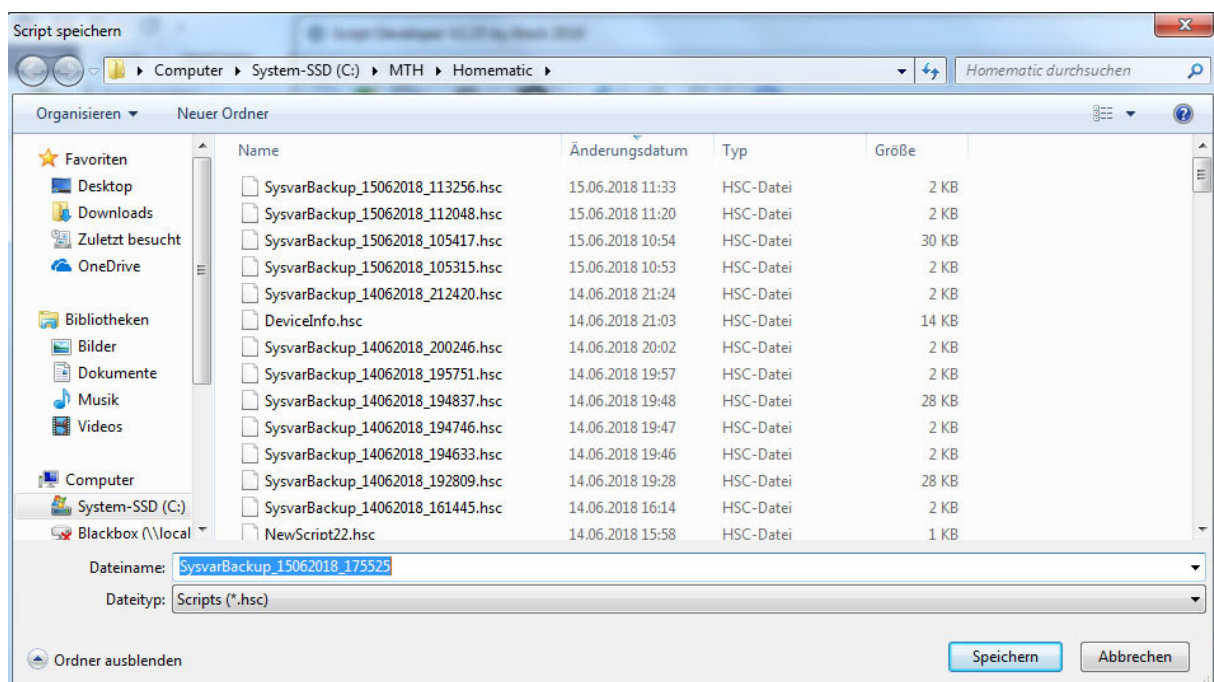
Selektierte Systemvariablen lassen sich durch Druck auf den Button „Lösche selektierte Sysvars“ nach Bestätigung der aufpoppenden Sicherheitsabfrage auf der CCU löschen. Achtung, löschen, nur wiederherstellbar, wenn vorher Systembackup auf der CCU gemacht wurde. Hilfreich, wenn irgendein Programm Mengen von Systemvariablen erzeugt hat, die sich durch die oben beschriebenen Filtermechanismen eingrenzen und dann selektieren lassen. Durch den Button lassen sich diese dann auf einmal entfernen.

4.3.2 Backup Systemvariablen

SystemVariablen können gesichert werden. Entweder eine Selektion von zu sichernden SysVars (beispielsweise wenn man in der Entwicklungsphase eines Scriptes ist) oder alle Systemvariablen



Backup erzeugt automatisch ein CCU Programm, welches anschließend gespeichert werden kann.



Der automatisch vorgeschlagene Dateiname ist immer SysvarBackup_Datum_Uhrzeit. Das erzeugte und gespeicherte Programm lässt sich dann im Scripteditor laden und öffnen

```
!-          Backup SystemVariablen vom 15.06.2018 17:55:25
!-          Erstellt mit Script Developer V2.25 by Black 2018
!----- Diese Zeilen Anpassen -----
boolean bcreate= true; !- Anlegen, wenn noch nicht existierte
boolean bupdate= true; !- Wert Updaten, wenn vorhanden und gleicher Typ
boolean barchive= false; !- false: immer restore mit DPArchive (false), true: restore mit altem Wert
!-----
string sID; object oSV; object rSV= dom.GetObject (ID_SYSTEM_VARIABLES);string svName; object oCHN;
string neu=' neu angelegt'; string alt=' existiert schon';
!-----
svName= 'KommErr.Heute'; oSV=rSV.Get (svName);
if (oSV) {
    if ((oSV.ValueType()==4) && (oSV.ValueSubType()==0) && bupdate) {
        WriteLine (svName # alt);
        oSV.State (2.000000);}
} elseif (bcreate) {
    oSV = dom.CreateObject (1089); rSV.Add (oSV.ID()); oSV.Name (svName); oSV.ValueType (4);
oSV.ValueSubType (0);
    oSV.DPInfo ('Heute Kommunikationsstörungen'); oSV.ValueUnit (''); oSV.Internal (false);
oSV.Visible (true);
    oSV.DPArchive (false && barchive);
    oSV.State (2.000000); oSV.ValueMin (0); oSV.ValueMax (100000);
    WriteLine (svName # neu);dom.RTUpdate(false);}
!-----
svName= 'KommErr.LastDevice'; oSV=rSV.Get (svName);
if (oSV) {
    if ((oSV.ValueType()==20) && (oSV.ValueSubType()==11) && bupdate) {
        WriteLine (svName # alt);
        oSV.State ('HM-LC-Sw2-FM LEQ1319211');}
} elseif (bcreate) {
    oSV = dom.CreateObject (1089); rSV.Add (oSV.ID()); oSV.Name (svName); oSV.ValueType (20);
oSV.ValueSubType (11);
    oSV.DPInfo ('Kommunikationsstörungen '); oSV.ValueUnit (''); oSV.Internal (false); oSV.Visible
(true);
    oSV.DPArchive (false && barchive);
    oSV.State ('HM-LC-Sw2-FM LEQ1319211');
    WriteLine (svName # neu);dom.RTUpdate(false);}
```

Für die zwei Systemvariablen wird folgendes Programm erzeugt.

Wichtig ist der Programmkopf:

Dort steht als Info im Kommentar das Erzeugerdatum und Zeit des Scriptes.

Ebenso folgende Angaben:

boolean bcreate= true; !- Anlegen, wenn noch nicht existierte

bei true wird eine Systemvariable, deren Namen noch nicht existiert, neu angelegt. Bei false wird NICHT neu angelegt.

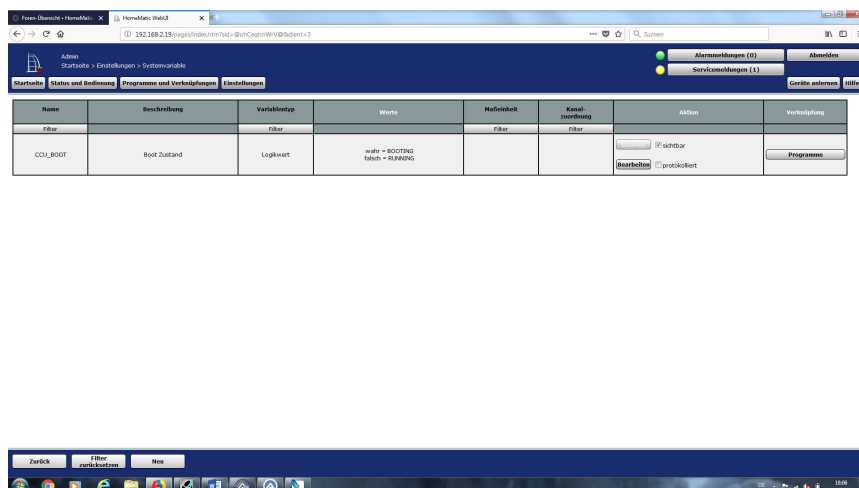
boolean bupdate= true; !- Wert Updaten, wenn vorhanden und gleicher Typ

Existiert schon eine Sysvar auf der CCU mit gleichem Typ (und gleichem Subtyp), dann wird bei bupdate=true die Systemvariable auf den Wert des Updatescriptes gesetzt, bei false wird eine bestehende Systemvariable nicht angefasst

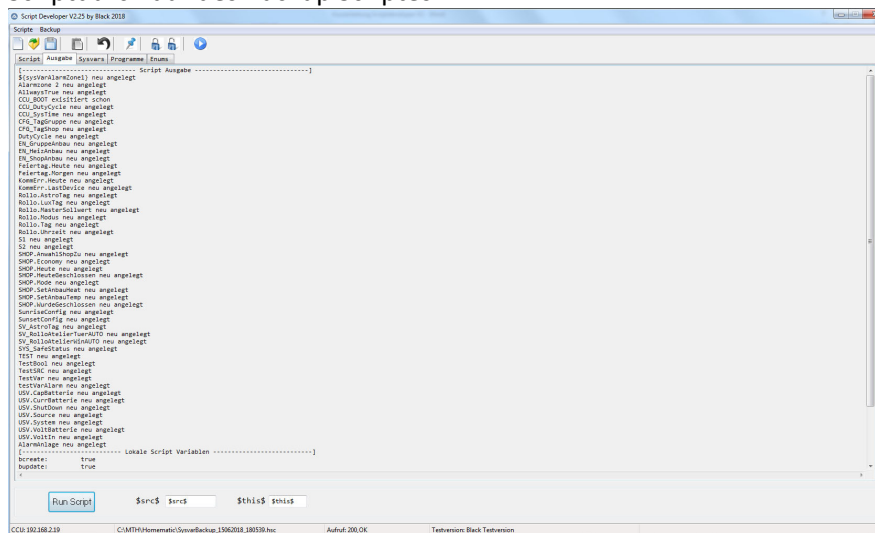
boolean barchive= false; !- false: immer restore mit DPArchive (false), true: restore mit altem Wert

Bei barchive=False wird das ProtokolierFlag der Sysvar auf false gesetzt, wenn diese neu angelegt wird, bei true wird der Wert des ProtokolierFlags aus dem Backup genommen. Feature beim Umzug eines Systems und Neuanlage, um alte Protokoliereinträge nicht ins neue System zu übernehmen

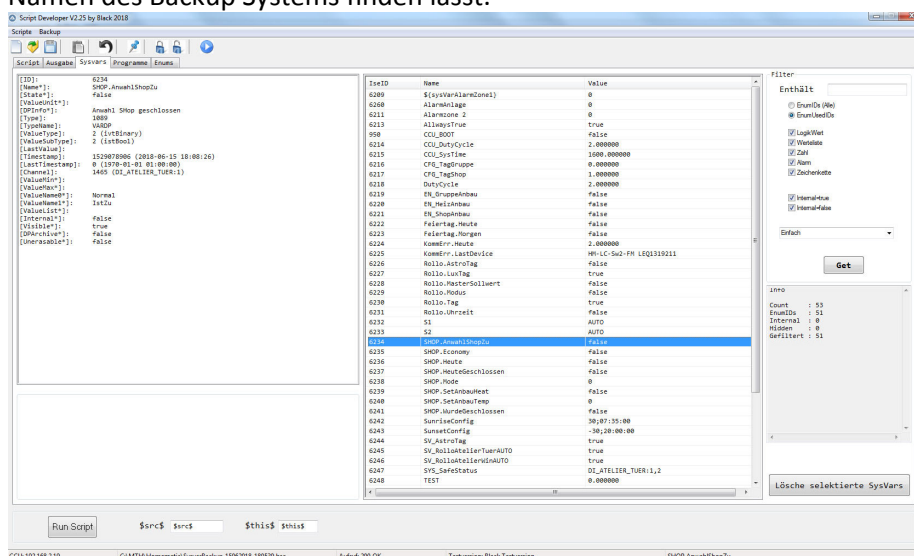
Gelöschte Systemvariablen auf meiner Test-Raspberrymatic



Scriptdurchlauf des Backup Scripts



Danach waren die Systemvariablen in korrekter Anzahl und Werten wieder da. Auch zugewiesene Kanäle werden rekonstruiert, so sich auf dem Restore System ein Kanal mit dem Namen des Backup Systems finden lässt.



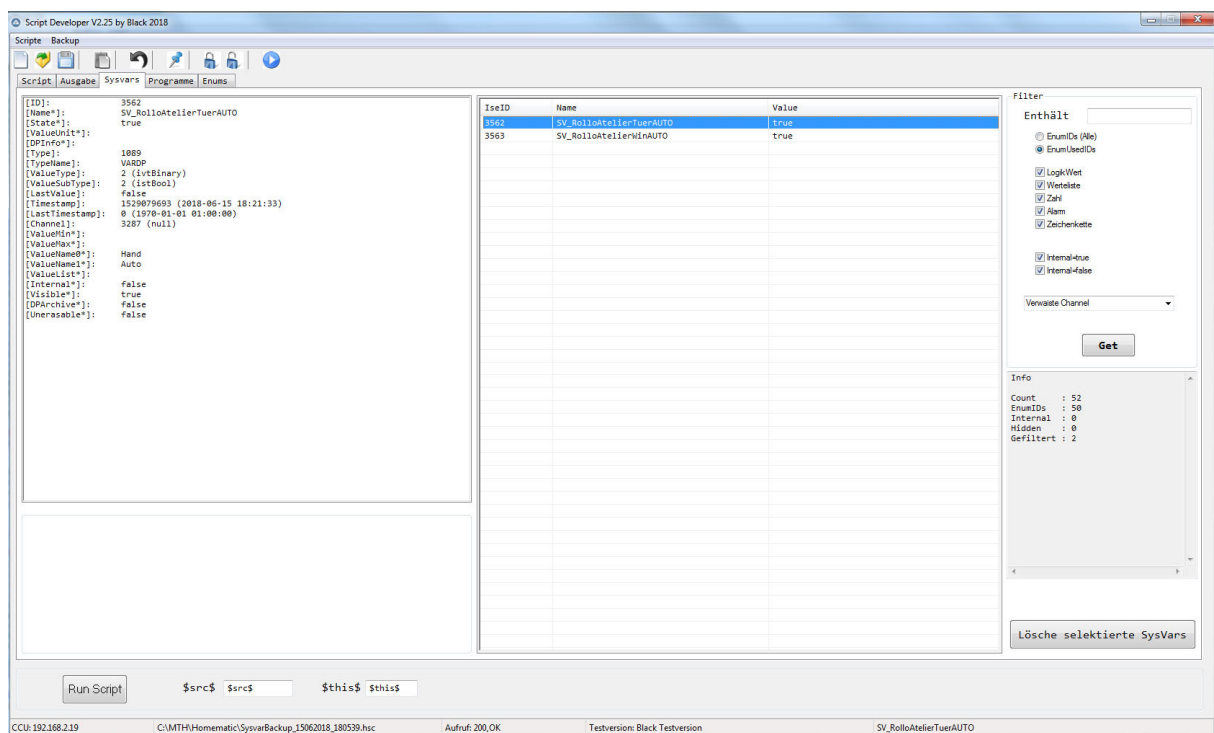
4.3.3 Spezielle Filter

4.3.3.1 Verwaiste Channel

Beim Programmieren des Script Developers und dem Test der Backup / Restore Routine fiel mir ein Fehler auf, das manche Systemvariablen, welche früher noch einem Kanal zugewiesen waren, dieses Gerät aber schon lange gelöscht war, immer noch die alte Kanal ID unter Channel stehen hatte. Dazu gibt's nun den Filter „verwaiste Channel“. Er findet Systemvariablen mit einer Channel ID, die aber nicht mehr auf einen gültigen Kanal zeigen. Dies kann bei Bedarf dann händisch korrigiert werden.

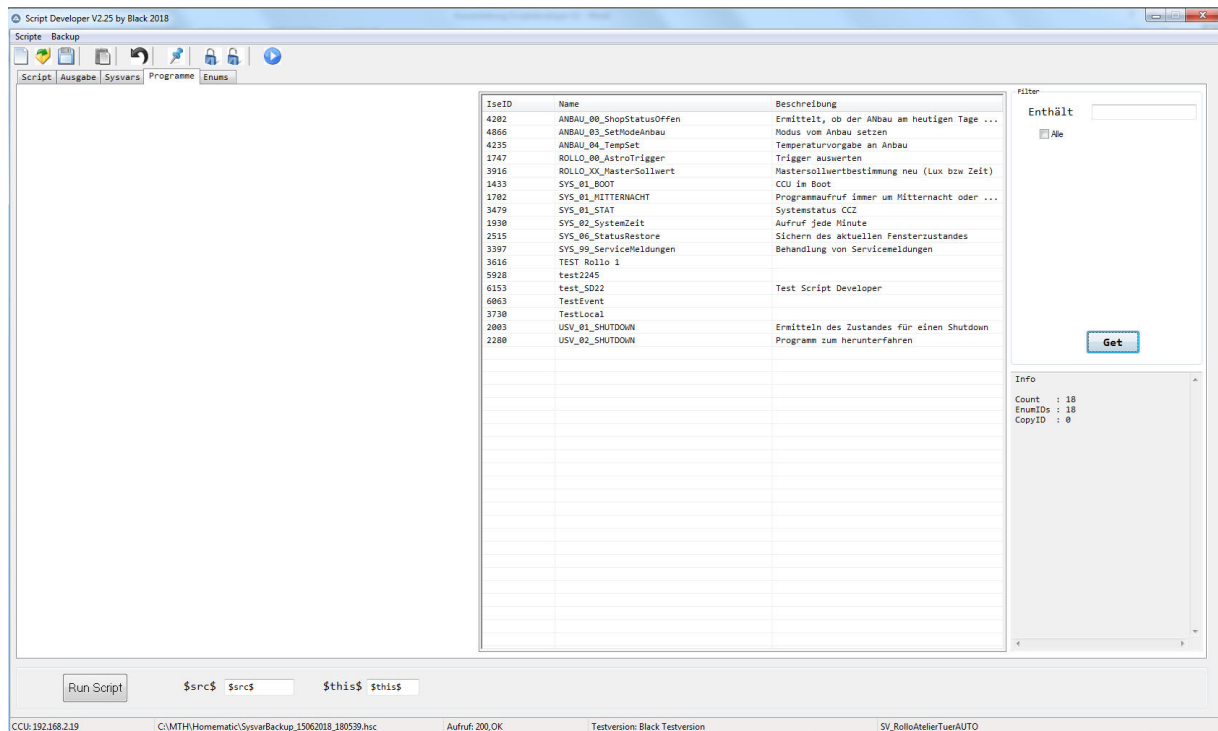
(Mehrfach Selektion als ENUMLIST erzeugen, unter Scripten einfügen erzeugt vordefinierte Schleife, dort dann manuell oSV.Channel (65535); würde das Problem eliminieren.

Für Ideen weiterer spezieller Filtermöglichkeiten, die man im Alltag brauchen kann, bin ich offen und würde diese auch implementieren. Hier zu sehen, Channel hat eine ID, wird aber als null (kein gültiges Object) erkannt



5. Programme

Der Reiter ist noch in Entwicklung, wird in einem der nächsten Updates mit Leben gefüllt sein.



Programme können schon dargestellt werden.

Hilfreich ist das Info Fenster.

Die Zahl CopIDs sollte 0 sein, (wenn man nicht gerade zeitgleich in der ClickiBunti ein Programm editiert). Eine Zahl ungleich 0 ist ein Zeichen für eine GeisterProgrammLeiche.

6. Scripterstellung mit Notepad++

Notepad++ ist ein geeigneter Editor, um mit Syntax Highlighting etc Script zu erstellen. Ebenso verfügt der Editor über Zeilennummern. Leider Optionen, die der Script Developer wohl nicht bekommen wird, dafür eine UDF zu programmieren übersteigt den Nutzen in meinen Augen.

Im Verzeichnis des Script Developers ist die neueste Version der Homematik Sprachbeschreibung . dieses unter eigene Sprachen importieren und schon hat Notepad++ auch Syntax Highlights für die CCU Scripterstellung.

Das Gute an Notepad ist, es kann unter dem Menüpunkt : Ausführen: externes programm ausführen ein programm starten, an welches auch noch weitere Parameter übergeben werden.
hier setzte ich dann mit meinem Script Developer an. der Aufruf des Developers geschieht dann über das Kommando `C:\PFAD_VON_CCUHTTP\ccuhttp.exe "$(FULL_CURRENT_PATH)"`

nach der Übergabe aus Notepad öffnet sich dann der Developer mit dem übergeben Script