


Kurzanleitung Scriptdeveloper V3.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 BetaTestversion.

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 V3.xx.xx	1
1. Installation	3
1.1 Lizenzsierung	4
1.2 Systemvoraussetzungen	7
1.3 Was tuts bis jetzt	7
1.4 Bekannte Einschränkungen / Bugs	7
2 Oberfläche	8
3 Scripteditor	9
3.1 Voreinstellungen Editor	10
3.2 Vervollständigen Funktion	11
4 Systemvariablen	12
5 Aufzählungen (Enums)	13

1. Installation

Das *.rar File in ein beliebiges Verzeichnis entpacken. Ein Installer ist nicht notwendig. In diesem Verzeichnis befindet sich auch das Konfigurationsfile SDV.INI. Bei der erstmaligen Verwendung muss dieses angepasst werden

[LAST]

DATEI=c:\MTH\Homematic\NewScript.hsc

[HOST]

CCU=192.168.2.19

NICKNAME=Benutzername

← anpassen Benutzername / Nick

CCU1=192.168.2.XX

← IP Adresser der 1. CCU

CCU2=192.168.2.XX

← Wenn vorhanden, IP der 2. CCU

CUXD=CUXD.CUX2801001:5

← CUXd KANAL (ist nötig, dafür braucht es kein pscp mehr)

LICENCE1=

← Lizenzschlüssel für 1. CCU

LICENCE2=

← Lizenzschlüssel für 2. CCU

[ENUM_NORM]

← Ab hier kommen dann interne Werte, Finger Weg

C1=65

C2=200

C3=293

C4=65

[ENUM_MAX]

C1=65

C2=200

C3=293

C4=65

Warum CUXD ? der Scriptdeveloper Version 2.x nutzte noch pscp für den Zugriff auf die Logdatei und auf das System. Dies war immer ein Schwachpunkt (zusätzliches Programm, Bestätigung Serverzertifikat. Dies wird jetzt mit CUXD realisiert. Es muss ein Kanal angegeben werden auf einem CUXD exec Gerät, auf das der Script Developer exklusiven Zugriff hat. Auf Systemen ohne CUXd kann der Script Developer nicht eingesetzt werden.

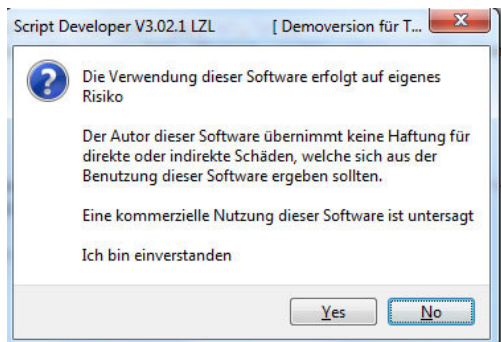
1.1 Lizenzierung

Der Scriptdeveloper ist bei nicht kommerzieller Nutzung Freeware. Trotzdem habe ich mich entschlossen, aufgrund von Erfahrungen der Vergangenheit den Nutzerkreis oder die möglichen Features bestimmter Nutzer einzugrenzen. Dies geschieht durch Vergabe von bis zu 2 Lizenzschlüsseln. Der Scriptdeveloper ist dadurch an bis zu 2 CCU / raspberrymatic gepairt.

Wie arbeitet das ?

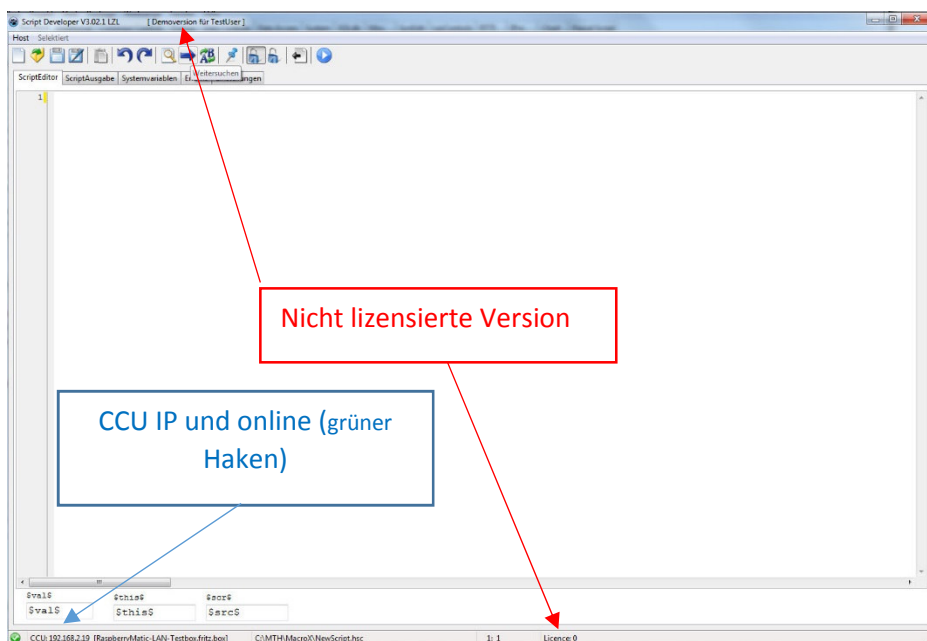
Der Script Developer telefoniert nicht nach Hause. Um eine Lizenz anzufragen ist folgender Weg einzuschlagen.

1. Die Konfigurationsdatei SDV.INI mit einem Editor öffnen.
2. Nickname Anpassen
3. IP der CCU 1 eintragen
4. IP der CCU 2 eintragen
5. CUXD Kanal eintragen
6. Konfiguration abspeichern
7. Script Developer starten

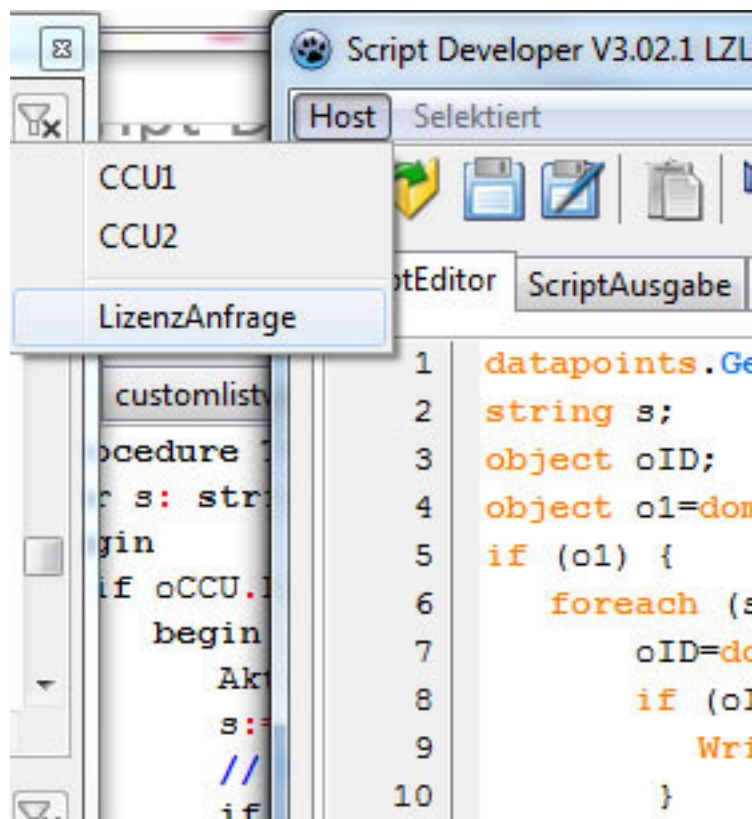


Bei allerersten Start muss dieses Fenster mit yes bestätigt werden. No führt so einem sofortigen Programmabbruch

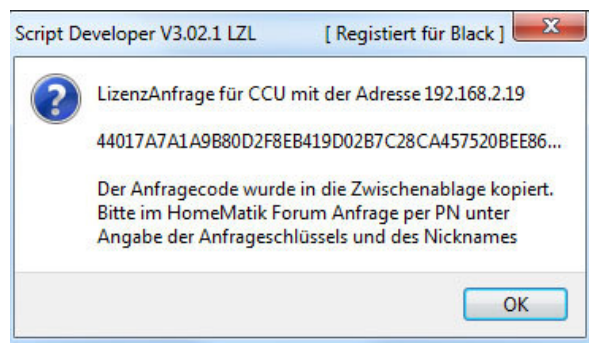
Bei Bestätigung mit Yes startet nun zum erstenmal der Scriptdeveloper als Demoversion



Für die weiteren Schritte muss der Script Developer mit der CCU verbunden sein und die CCU auch als online erkannt worden sein.



unter Host auf Lizenzanfrage drücken. Als nächstes öffnet sich ein Fenster mit einem Anfrage Hexstring.



Der Hexstring ist in die Zwischenanlage kopiert und kann in beliebige Text Dokumente eingefügt werden. Als nächste dann im Homematik.de Forum eine PN an mich schreiben mit dem String und Angabe des Nicknames, welcher zum Zeitpunkt der Lizenzanfrage in der INI Datei eingetragen war.

Was enthält dieser Hexstring ?

Kodiert und verschlüsselt: 1. den Nicknamen, 2. die Seriennummer des Funkmodules der verbundenen CCU , einen VerifizierCode von mir.

Die Seriennummer des Funkmodules ist nötig zur Verifizierung des Pairings. Diese wird bei mir nirgends gespeichert, mit diesem Hexschlüssel wird nach der Anfrage der LizenzLevel definiert und ebenfalls in einen Hexstring kopiert. Dieser dann zurückgesandte Hexstring wird unter Licence1 oder Licence2 in der INI Datei eingetragen. Es kann mit bis zu 2 CCU bearbeitet werden, sollte ein Lizenzlevel höherwertiger sein so gilt dieser höherwertige Level für beide CCUs.

Wer mit diesem Verfahren nicht einverstanden ist, möge bitte an dieser Stelle die PDF Datei schließen und kann die Dateien beruhigt löschen.

Geplant hab ich folgende Lizenzabstufungen

Level	Editor	Script ausführen	Highlighter Und Vervollständiger	Enums	SysVar	Programs	Backup Restore		
0	X								
1	X	X							
2	X	X	X						
3	X	X	X	X	X				
4	X	X	X	X	X	X			
5	X	X	X	X	X	X	X		
6									

1.2 Systemvoraussetzungen

Der Scriptdeveloper lief bisher in Testinstallationen unter WIN 7 64/32 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 3.37.8.20181026 getestet.

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

1.3 Was tuts bis jetzt

Der Editor funktioniert inkl. Suchen und Suchen / ersetzen. Der Highlighter und der Code vervollständiger arbeiten auch.

Undo / Redo arbeiten

Script ausführen arbeitet und liefert wie in der alten Version die Antworten der CCU.

Enums arbeiten auch schon inkl. Detaildaten und Editiermöglichkeiten.

1.4 Bekannte Einschränkungen / Bugs

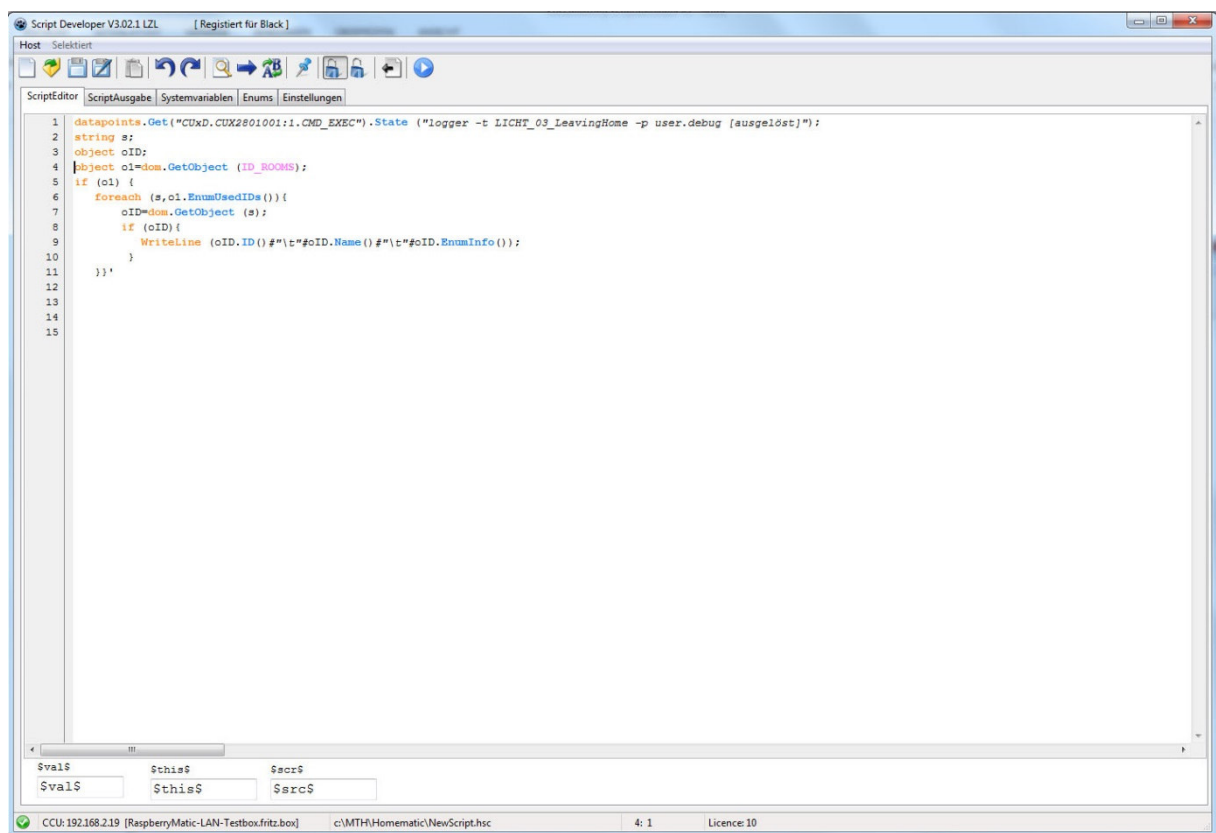
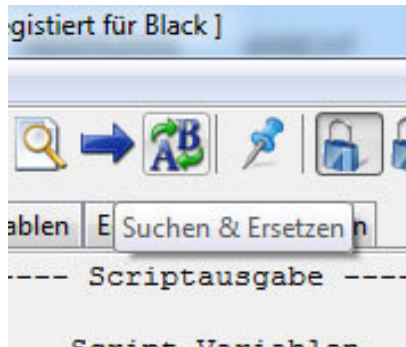
Auswahldialoge sind auf Englisch. Weiß ich, zurzeit benutze ich die in der Laufzeitumgebung integrierten Dialoge, und die sind leider trotz Landeseinstellung englisch.




Folding im Editor arbeitet noch nicht. Wenn der Rest läuft gucke ich da mal nach.

Kommentare im Script müssen als !- geschrieben werden. Kann man sich dran gewöhnen, das anzupassen wäre ein Haufen Aufwand, da EQ3 ja klugerweise Negation und Kommentar mit demselben Zeichen bedacht hat. Hurra. Ich kann jedenfalls mit dem !- gut leben, folglich ist die Chance, das ich das ändere, recht gering: xD

2 Oberfläche

Zu fast allen Funktionen sind die Hint paramtriert, so das es da Hilfestellung gibt.




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. Clipboard Inhalt (dazu später mehr, wenns programmiert ist)

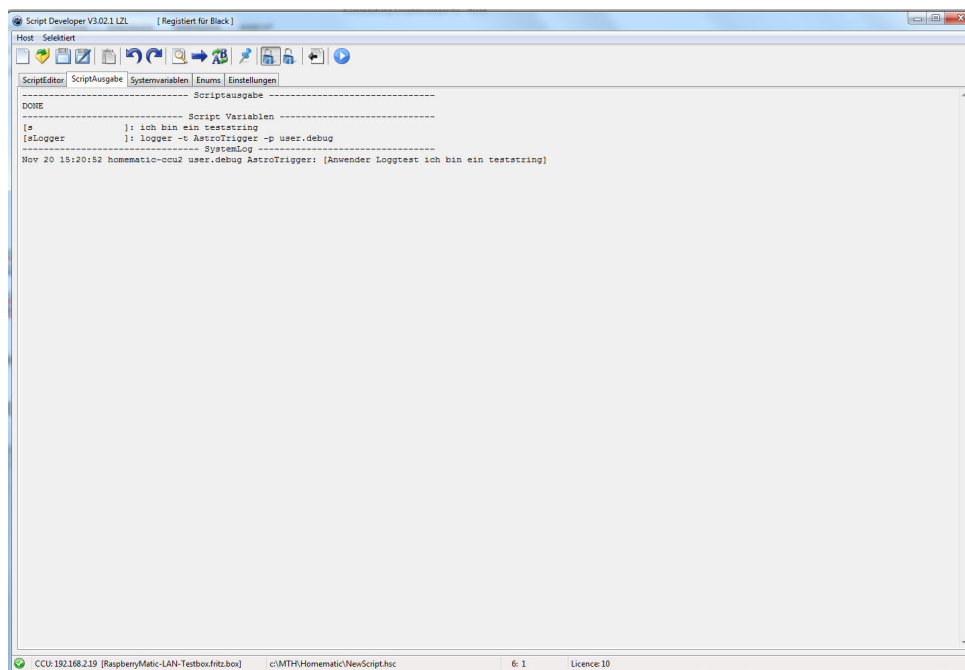
3 Scripteditor

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 testweise aus State

Erzeugt folgende Ausgabe:



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

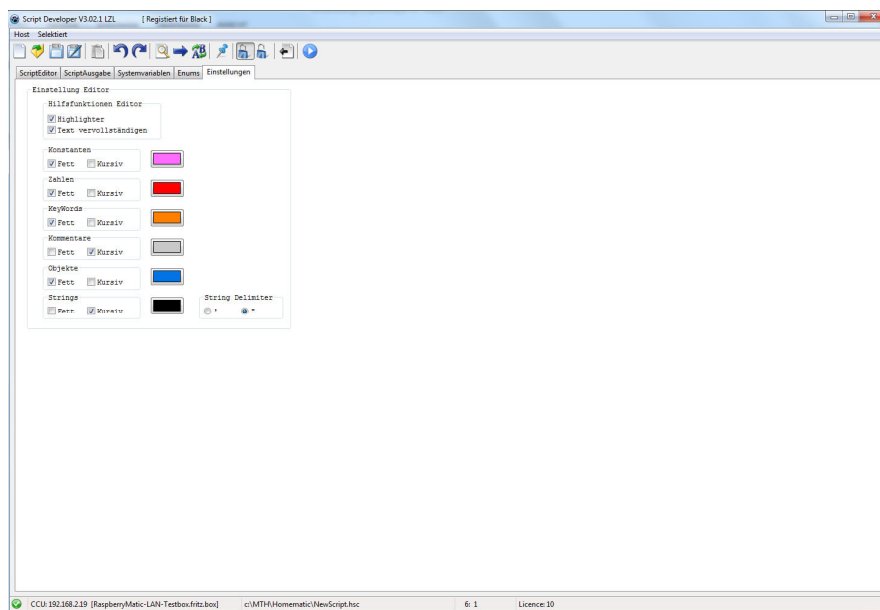
Unter lokale Script variablen stehen die Variablen welche im Script definiert wurden mit ihren Namen. In dem Fall hier sind das die Beiden String Variablen s und sLogger.

Wurde via Userlog ein Eintrag im Logfile erzeugt, so wird dieser nach Scriptende auch hier angezeigt.

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]
```

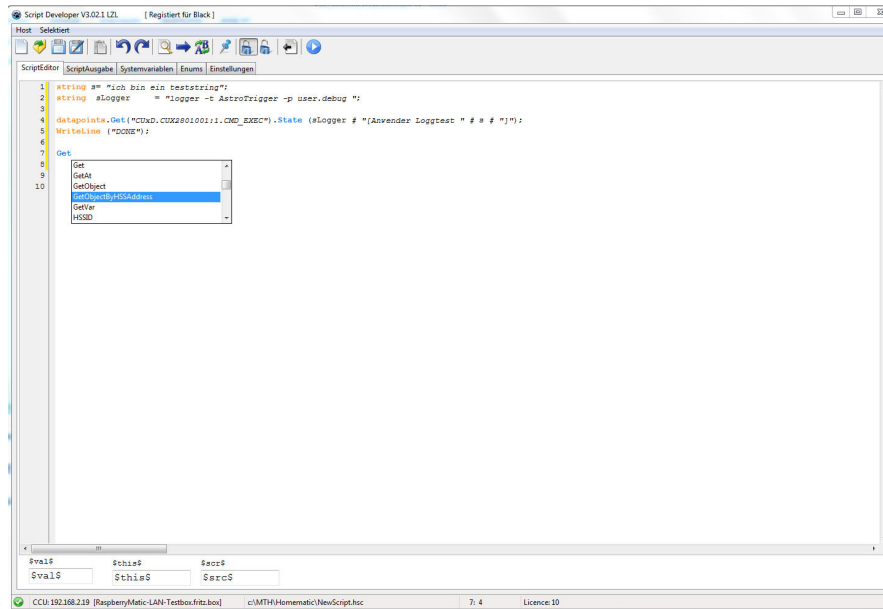
3.1 Voreinstellungen Editor



Hier kann nach Vorliebe der Highlighter konfiguriert oder auch ausgeschaltet werden.
Ebenso lässt sich die Vervollständigen Funktion an oder abwählen.
Vorbereitung natürlich: Lizenzlevel muss vorhanden sein.

3.2 Vervollständigen Funktion

Methoden und Konstanten Namen muss man sich nicht auswendig merken. Der Editor verfügt über einen Auto Vervollständiger. Man schreibt den Wortanfang, hier z.B Get , drückt Strg+Space und wählt in dem sich öffnenden Menü die passende Funktion aus.

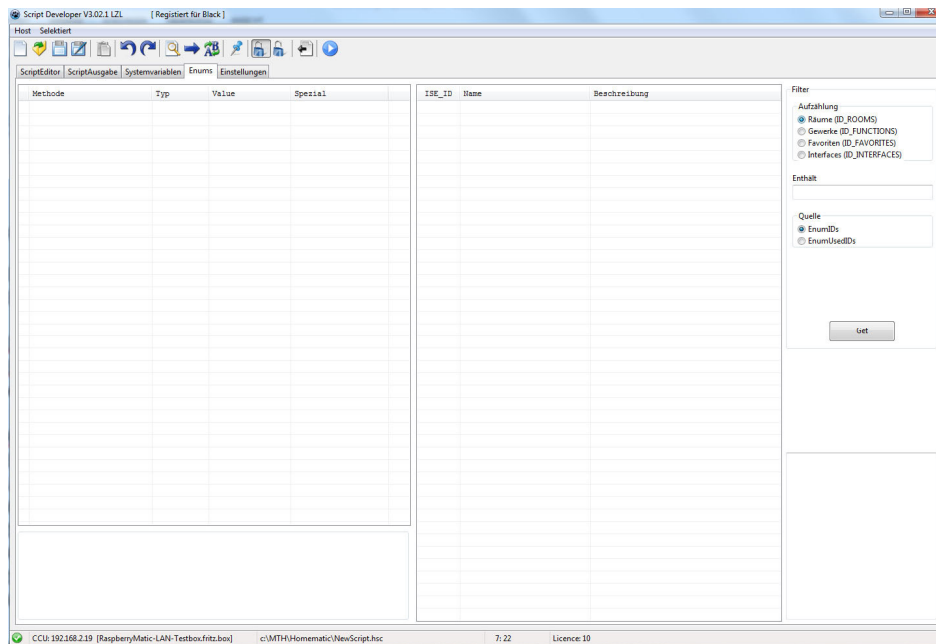


Nach Druck von Enter erscheint das Wort im Editor.

4 Systemvariablen

Kommen in einer folgenden Version

5 Aufzählungen (Enums)



Filteroptionen:

Auswahl der Aufzählungen (Räume, Gewerke, Favoriten, Interfaces sind bisher implementiert)

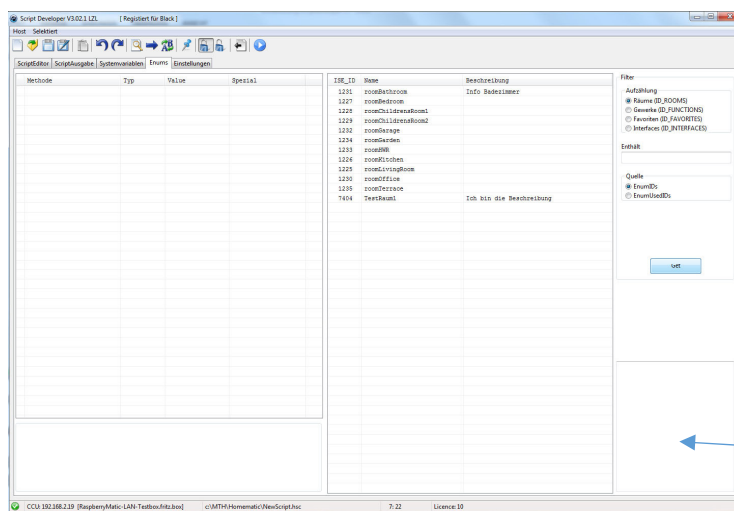
Enthält: leerer Eintrag = es wird nicht nach enthaltener Buchstabensequenz selektiert
Eingegebener Text. Die Systemvariable muss im Namen die Buchstabensequenz enthalten.

EnumIDs : es werden alle Systemvariablen durchsucht

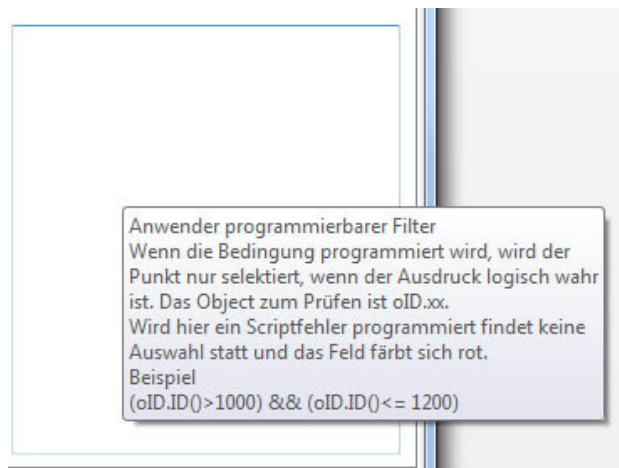
EnumUsedIDs: es werden nur die Systemvariablen durchsucht, welche auch in der WebUI dargestellt werden

Durch Druck auf Get wird die Liste gemäß Selektion von der CCU angefordert, aufbereitet und dargestellt. (Lizenzlevel vorausgesetzt)

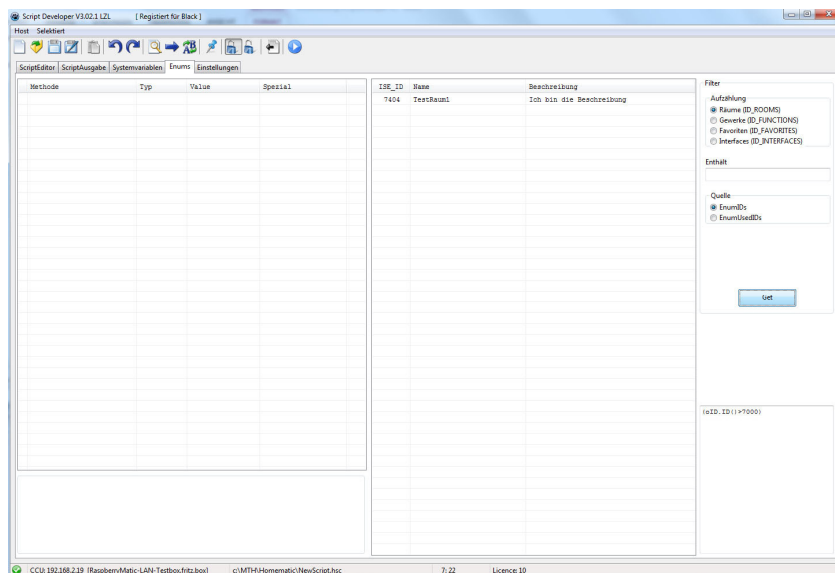
Beispiel Räume:



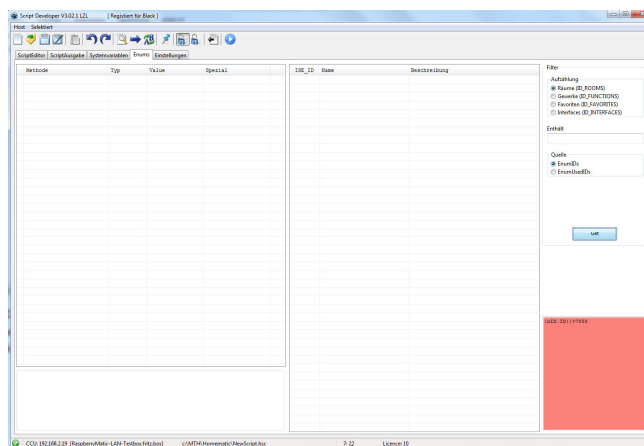
Zusätzliches Feld für
Selektionsbedingen



Beispiel hier für einen ganz einfachen Filter, stelle nur Räume dar, deren lseID grösser 7000 ist:



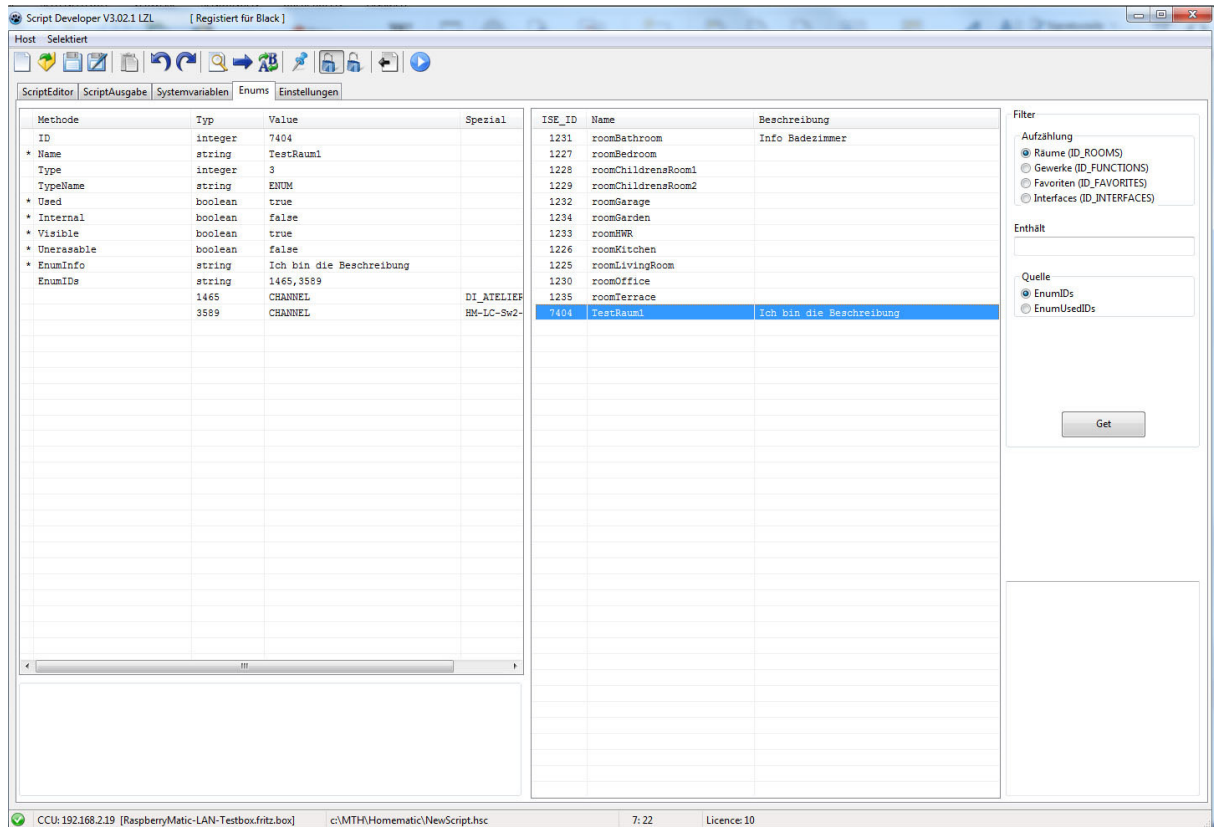
Die Filterbedingung wird in HM Script ausformuliert. Das gefundene Object kommt nur in die Liste, wenn die ausformulierte Bedingung True ist. Das Teil ist mächtig, aber auch nicht ungefährlich, man kann auch Müll als Bedingung schreiben. Dabei kommt dann aber eine Warnung:



Bedingung ist falsch: erzeugt Scriptfehler (Klammer zu fehlt). In dem fall färbt sich nach Druck auf Get das Feld rot

Durch Click auf die Beschreibungszeile IseID bzw Name können die Felder entsprechend sortiert werden.

Click auf eine selektierte Aufzählung öffnet im Detailfenster die Methodenansicht des Objektes



Die Spaltenbreite kann sowohl in Normalansicht als auch in Maximize separat eingestellt werden (Das Programm sollte sich die Breiten merken und je nach Darstellungsart automatisch wieder einstellen, sollte...)

Dargestellt werden die Methode, der Vartype und die Property.

Bei den Aufzählungen wird jeweils eine Rekursionsstufe aufgelöst, um an die Detailinformationen zu kommen. Hier die Liste der Channels, die diesen Raum verwenden, aufgelöst in die ID, der Typ (hier Channels und der Name des Channels)

Dazu auf die Zeile klicken

