

## 1 Vorbemerkung

Im ersten Anlauf ging es mir zunächst einmal nur darum, die .json Dateien etwas lesbarer aufzubereiten. Dafür habe ich ein kleines Programm erstellt, welches eine .json Datei einliest, Tabulatorsprünge und Zeilenvorschübe entsprechend den in der Datei vorhandenen geschweiften Klammern einfügt, und das Ergebnis dann unter neuem Namen mit Endung .txt statt .json in eine neue Datei schreibt. Diese Textdatei kann mit jedem beliebigen Editor geöffnet werden. Der enthaltene Text wird dann in strukturierter Form dargestellt.

Durch die vielen Parameter ist das Ganze aber trotzdem noch relativ unübersichtlich. Das hat mich dann auf die Idee gebracht, die für mich interessanten Parameter aus der Datei herauszuziehen und in einer Excel Tabelle aufzubereiten. Ich habe mich deshalb für Excel entschieden, weil ich davon ausgegangen bin, dass ich mir relativ viel Programmieraufwand sparen kann, wenn ich die Möglichkeiten, die Excel mit der Verwendung von Formeln bietet, nutzen kann.

Ziel war nun, ein einigermaßen flexibles Werkzeug zu schaffen, so dass die verschiedenen Parameter mit möglichst allgemeingültigen Programmerroutinen oder einfachen Formeln so aufbereitet werden, dass die wichtigsten Informationen auf einen Blick zur Verfügung stehen.

## 2 Tabellenaufbau

Die Excel Datei enthält 2 verschiedene Tabellenblätter. Die eigentlichen Modelldaten sind im Tabellenblatt **Modellwerte** hinterlegt. Das Tabellenblatt **Umsetzungstabellen** enthält diverse Umsetzungstabellen für die Umsetzung der originalen Parameterwerte in leichter lesbare Informationen.

Die Tabelle im Tabellenblatt **Modellwerte** besteht im wesentlichen aus 2 verschiedenen Teilen:

### 2.1 Liste von Mono-Parametern

Als Mono-Parameter habe ich diejenigen Parameter bezeichnet, die es nur einmal in der Datei gibt und die auch nur einen Parameterwert haben. Als Beispiel sei hier der Lehrer-Schüler-Schalter genannt.

Die Liste der Mono-Parameter steht ganz am Anfang der Datei, sie beinhaltet die Spalten 1 bis 5 wobei die Spalten 2 bis 5 normalerweise ausgeblendet sind. Die Spalten sind wie folgt belegt:

- a) enthält eine frei vergebene Parameterbezeichnung
- b) enthält die genaue Zeichenkette, mit der der Parameter in der Modelldatei gesucht wird
- c) enthält den Parametertyp „mono“
- d) enthält den in der Modelldatei gefundenen Parameterwert
- e) enthält den aus Spalte 4 übernommenen und erforderlichenfalls lesbar aufbereiteten Parameterwert

### 2.2 Liste von Multi-Parametern

Als Multi-Parameter habe ich diejenigen Parameter bezeichnet, die mehrfach in der Datei vorkommen und die mehrere Parameterwerte haben. Ein typischer Vertreter sind z.B. die

Servoeinstellungen. Die Spalten in diesem Teil der Tabelle sind unterschiedlich belegt, je nachdem ob es sich um eine Zeile mit Überschriften handelt oder ob eine Zeile Parameterwerte enthält:

Die Spalten der Überschriftszeile sind wie folgt belegt:

- a) w.o.
- b) w.o
- c) enthält den Parametertyp „multi“ plus Zusatzinfos
- d) nicht genutzt
- e) enthält eine Beschreibung der nachfolgenden Aufbereitung
- f) und folgende enthalten die originalen Parameterbezeichnungen aus der Modelldatei

Die Spalten der Datenzeilen der Multi-Parameter sind wie folgt belegt:

- a) **muss** das Zeichen „>“ enthalten
- b) nicht genutzt, kann für Umsetzungsformeln verwendet werden (Derzeit nur bei Mixern genutzt)
- c) nicht genutzt, kann für Umsetzungsformeln verwendet werden
- d) nicht genutzt, kann für Umsetzungsformeln verwendet werden
- e) enthält eine aus den Spalten f und folgende zusammengesetzte, lesbare Aufbereitung unterschiedlicher Parameter. Die Spaltenüberschriften derjenigen Spalten, die zur Aufbereitung herangezogen werden, sind mit grünem Hintergrund versehen. Außerdem enthalten sie einen Kommentar zur Art und Weise der Aufbereitung.
- f) und folgende enthalten die originalen Parameterwerte aus der Modelldatei.

## 3 Nutzung der Exceldatei

### 3.1 Allgemeines

Damit das Tabellenblatt mit den Informationen aus einer beliebigen .json Modelldatei genutzt werden kann, muss die Makroausführung zugelassen sein, da sowohl Makros als auch selbst erstellte Funktionen genutzt werden. Das Befüllen der Tabelle erfolgt über einen Command-Button im Tabellenblatt **Modellwerte**.

Nach Drücken des Buttons wird über ein VBA-Makro zunächst die zu verarbeitende Datei abgefragt und wie in Ziff. 1 beschrieben in eine .txt Datei konvertiert. Dies wurde bewusst so beibehalten, da die Fehleranalyse bei ev. auftretenden Problemen mit der .txt Datei etwas einfacher wird. Die Datei wird in das Verzeichnis geschrieben, in dem auch die .json Datei steht. Es empfiehlt sich daher, die .json Dateien vorher vom Sender auf den PC zu übertragen.

Anschließend wird erste Spalte abgearbeitet. Die dort festgelegten Parameter werden gesucht und deren Werte in die Tabelle übernommen. Von Jeti grundsätzlich benutzte Platzhalter für bestimmte Multiparameter werden bis auf die Überschriftszeilen und die erste Parameterzeile nach der Abarbeitung der Eingabedatei gelöscht, sofern sie nicht wirklich genutzte Werte enthalten. Die erste Wertezeile bleibt jedoch immer erhalten und dient als Vorlage für den Aufbau neuer Zeilen zum jeweiligen Multi-Parameter. Wird auch diese nicht genutzt, wird sie samt Überschriftszeile ausgeblendet. So wird gewährleistet, dass in den Wertezeilen in den Spalten 2 bis 5 enthaltene Formeln nicht verloren gehen.

### 3.2 Normale Nutzung

Bei „normaler“ Nutzung wird eigentlich nur der o.g. Command-Button benötigt. Es sollten keine Zeilen gelöscht und keine Formeln geändert werden. Da das Tabellenblatt bei eventuellen Fehlern teilweise überschrieben werden kann, empfiehlt es sich eine Sicherungskopie unter anderem Namen zu halten. Eine andere Variante wäre, aus der .xlsm Datei eine Excelvorlage (.xlt oder xltx) zu erstellen und dann jeweils eine neue Exceldatei über „Datei neu“ und Auswahl der Vorlage zu erstellen. Das hätte dann auch den Vorteil, dass man für jedes Modell auch eine eigene Exceldatei hätte. Wird das nicht so gemacht, empfiehlt sich unter Umständen ein „speichern unter“ (ggf. im PDF Format).

**Wichtiger Hinweis:** *Sämtliche Makros, benutzerdefinierten Funktionen und die verwendeten Makros basieren auf der Sender-Firmware 3.02. Bei älteren Versionen sind Ungereimtheiten nicht auszuschließen.*

### 3.3 Erweiterte Nutzung

Sollen andere als die bisher ausgewählten Parameter hinzugefügt oder die Infos in der Spalte E erweitert werden, so ist das natürlich möglich, erfolgt aber auf eigenes Risiko. Bei den Multi-Parametern ist es wichtig, den richtigen Parametertyp anzugeben. Momentan sind nur die Besonderheiten der bisher genutzten Parameter abgefangen. Damit dürften schätzungsweise 80% der übrigen Parameter auch abgedeckt sein, es ist jedoch nicht auszuschließen, dass es noch weitere Besonderheiten gibt, die dann eine Anpassung des Bearbeitungsmakros erfordern würde.

Um die Formeln in der Spalte E so einfach wie möglich gestalten zu können wurden folgende eigene Funktionen erstellt:

1. sel\_par(woher,nr) zur Selektion eines bestimmten Stellungsparameters aus einer Zelle mit durch Komma getrennte Parameterwerte
2. z\_z(wert) zur Umrechnung der Zeit in Sekunden in das Excel Zeitformat
3. l\_sw(sw1,sw2,cond1,cond2,wert1,wert2,log\_typ) zur Aufbereitung der logischen Schalter aus den Werten der Parameterzeile
4. n\_sw(sw,cond,wert) zur Bestimmung eines Schalters inklusive Bedingungsparameter und vorgegebenem Wert
5. nur\_sw(sw) zur Bestimmung eines Schalters ohne Bedingung und Wert
6. zs\_zelle(zeile,spalte) zur Erstellung eines Zellwertes aufgrund einer Zeilen und Spaltenangabe
7. p\_wert(werk, anzahlkommastellen) zur Berechnung eines Parameterwertes in Abhängigkeit der Anzahl der Kommastellen

## 4 Schlussbemerkung

Wünsche viel Spaß mit und hoffentlich einen besseren Überblick zu allen Modelldefinition durch die Exceldatei. Sollten (wider Erwarten 😊) Probleme auftauchen helfe ich gerne via E-Mail an

[hermann.eichner@freent.de](mailto:hermann.eichner@freent.de)

weiter.