



# Dokumentation

# Tool4S7

## 1. Inhalt

1. Inhalt .....	1
2. Grundlegendes .....	2
3. Spracheinstellungen.....	3
4. Tools .....	3
4.1. Symbolgenerator.....	4
4.2. Quellengenerator- DB .....	5
4.3. Quellengenerator- Melde- DB/ UDT.....	5
4.4. Quellengenerator- DB- Abbild von Datei .....	6
4.5. Quellengenerator- AWL- Schrittkette- Simple .....	6
4.6. Quellengenerator- AWL- Schrittkette- Extended.....	7
4.7. Adressrechner .....	7
4.8. Pointer- Tool.....	7
5. Lizenzrechte .....	8
6. Änderungen/ Erweiterungen.....	8



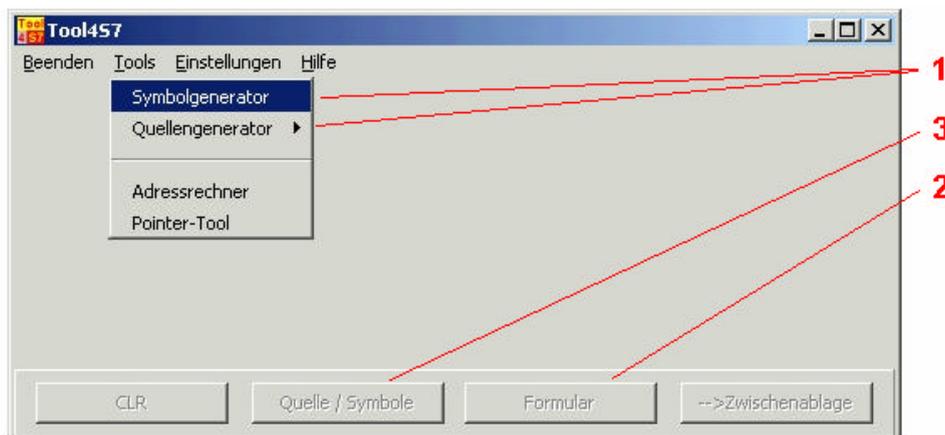
## 2. Grundlegendes

Tool4S7 enthält einige kleine Tools, welche dem S7-Programmierer das Leben etwas verschönern sollen.

Die Symbol- und Quellengeneratoren (1) besitzen im Prinzip 3 Editoren:

- Formular. Hier werden die Einstellungen für den entsprechenden Generator getroffen, und dieser gestartet.
- Unter Symboltabelle findet man evtl. generierte Symbole. Diese kann man direkt per Zwischenablage in die Step7-Symboltabelle kopieren.
- Unter Quelle findet man den generierten Programm Quelltext.

Mit den Buttons „Formular“ (2) und „Quelle / Symbole“ (3) kann man zwischen den 3 Editoren wechseln.



### Allgemeines zur Quelle

In jedem Step7-Programmordner kann man einen Ordner Quellen und darin beliebige Quellen erstellen. Normalerweise wird der Ordner Quellen sogar automatisch erstellt.

So kann man vorgehen:

- leere AWL- Quelle anlegen (rechter Mausklick auf den Quellordner.. Kontextmenü verwenden).
- Quelle durch einen Doppelklick öffnen.
- Den im Tool4S7 erzeugten Quellcode in die leere AWL- Quelle einfügen.
- Bei Bedarf editieren.
- Quelle übersetzen (im Step7-Editor Datei.. Übersetzen). Damit wird ein oder mehrere neue S7-Bausteine erzeugt.

Wenn der Quellcode keine Variablendeklaration enthält kann man ggf. den Quelltext direkt in einen AWL- Baustein kopieren, und somit den Weg über die AWL- Quelle einsparen.

### Beispiel- Buttons

In einigen Formularen sind Beispiel- Buttons zu finden. Es empfiehlt sich, diese Buttons zu verwenden. Anschließend generieren und den entstandenen Programmcode anschauen.

So bekommt man schnell einen Überblick, über die Möglichkeiten des entsprechenden Tools.



# Tool4S7

Tool4S7\_de.pdf  
Version Doku 03.08

[holger.reither@web.de](mailto:holger.reither@web.de)  
<http://reither.ohost.de/tool4s7.htm>

## 3. Spracheinstellungen

In Step7 kann man sowohl mit englischer wie auch mit deutscher Mnemonik programmieren (Einstellbar im Step7-AWL-Editor unter Extras.. Einstellungen..).

Analog dazu kann man im Tool4S7 unter Einstellung.. Sprache.. zwischen englischer und deutscher Mnemonik wählen.

Weiterhin kann man hier die Landessprache für Tool4S7 setzen. Die bisher programmierten Tools stehen allerdings bisher fast ausschließlich in deutscher Sprache zur Verfügung!





## 4. Tools

### 4.1. Symbolgenerator

Mit dem Symbolgenerator kann eine Reihe von aufeinander folgenden Symbolen erzeugt werden. Diese Symbole können anschließend direkt in den Step7- Symboleditor eingefügt werden.

Zur Bezeichnung der Symbole und Kommentare kann beliebiger Text und die Variablen <A1>, <A2>, <C1>, <C2>, <C3>, <T1> und <T2> verwendet werden.

<A1>, <A2>, <C1> und <C2> werden mit jedem neuen Symbol inkrementiert bzw. dekrementiert. <C3> wird nach jeder vollen Textsequenz inkrementiert. Das heißt, immer wenn alle Texte <T1>bzw. <T2> einmal verwendet wurden, wird <C3> erhöht. Und die Textvergabe <T1> bzw. <T2> beginnt mit einer neuen Sequenz.

Zum testen einfach mit den Beispielwerten experimentieren!

Zur besseren Übersicht kann das Eingabeformular vereinfacht bzw. erweitert dargestellt werden.

„Beispiel Programmnummer“

M_PrgNr_Bit0	M	10.0	Bool	Programmnummer Bit 0
M_PrgNr_Bit1	M	10.1	Bool	Programmnummer Bit 1
M_PrgNr_Bit2	M	10.2	Bool	Programmnummer Bit 2
M_PrgNr_Bit3	M	10.3	Bool	Programmnummer Bit 3
M_PrgNr_Bit4	M	10.4	Bool	Programmnummer Bit 4
M_PrgNr_Bit5	M	10.5	Bool	Programmnummer Bit 5
M_PrgNr_Bit6	M	10.6	Bool	Programmnummer Bit 6
M_PrgNr_Bit7	M	10.7	Bool	Programmnummer Bit 7

„Beispiel Zylinder“

E50.0Zyl1ausg	E	50.0	Bool	Zylinder 1 ausgefahren
E50.1Zyl1eing	E	50.1	Bool	Zylinder 1 eingefahren
E50.2Zyl2ausg	E	50.2	Bool	Zylinder 2 ausgefahren
E50.3Zyl2eing	E	50.3	Bool	Zylinder 2 eingefahren
E50.4Zyl3ausg	E	50.4	Bool	Zylinder 3 ausgefahren
E50.5Zyl3eing	E	50.5	Bool	Zylinder 3 eingefahren
E50.6Zyl4ausg	E	50.6	Bool	Zylinder 4 ausgefahren
E50.7Zyl4eing	E	50.7	Bool	Zylinder 4 eingefahren
E51.0Zyl5ausg	E	51.0	Bool	Zylinder 5 ausgefahren
E51.1Zyl5eing	E	51.1	Bool	Zylinder 5 eingefahren



## 4.2. Quellengenerator- DB

Mit dem DB- Quellengenerator kann eine Reihe von aufeinander folgenden Variablen erzeugt werden. Eine Variable besteht aus einem Namens Kürzel gefolgt von einer Nummer oder Adresse. Dahinter kann ein sich abwechselnder Text (Sequenz) folgen. Z.B. „Vor“, „Aus“, „Rueck“. Adressen werden mit jeder neuen Variable inkrementiert. Nummern werden erst mit dem Beginn einer neuen Sequenz inkrementiert.

## 4.3. Quellengenerator- Melde- DB/ UDT

Ein kleines Problem von WinCC- Bitmeldungen ist, dass sie Word-orientiert sind. Damit erscheint die Anordnung der Meldebits im DB etwas chaotisch. Beim Programmieren der Meldebits kann man schnell durcheinander kommen. Deshalb wurde Tool Melde-DB/ UDT programmiert. Dieses Tool übernimmt die Anordnung der Meldebits im DB oder alternativ in einem UDT automatisch.

Variablenamen werden durchnummeriert und so angeordnet, dass sie im WinCC aufsteigend angeordnet erscheinen.

Betreffend der Bitmeldungen, ist die Verwendung von UDT's sehr empfehlenswert.

Zunächst erstellt man UDT und DB. Im Verlauf der Programmierung/ Inbetriebnahme können zu jeder Zeit die Meldetexte (als Kommentar) in den UDT eingetragen werden. Da es sich nur um Kommentare handelt, muss der Melde- DB nicht mehr auf der CPU aktualisiert werden. Trotzdem stehen die Meldetexte sofort im gesamten Projekt zur Verfügung. Regelmäßig die Kommentare in die WinCC- Bitmeldungen kopieren und man hat ein aufgeräumtes und gut kommentiertes Projekt, zumindest was die Bitmeldungen angeht.



## 4.4. Quellengenerator- DB- Abbild von Datei

Mit diesem Tool lässt sich eine Beliebige Datei in einen DB wandeln, so dass der Inhalt der Datei anschließend als Array in einem Datenbaustein verfügbar ist.

Arrays müssen mindestens 1 Element enthalten. Das größtmögliche Array beginnt theoretisch bei -32767 und endet bei 32767.

Alle möglichen Bytes von 00h bis FFh können auch als Char ausgegeben werden.

Nicht darstellbare Zeichen erhalten einen Ersatzcode.

Z.B. \$" , \$05 oder \$P .  
(Benannte Steuercodes:  
\$T .. \$09 .. Tabulator  
\$L .. \$0A .. Zeilenvorschub  
\$P .. \$0C .. Seitenvorschub  
\$R .. \$0D .. Wagenrücklauf )

## 4.5. Quellengenerator- AWL- Schrittkette- Simple

Hiermit kann das Gerüst für eine Schrittkette erzeugt werden. Benötigte Symbole werden parallel dazu erstellt.

Einige Eigenschaften des erzeugten Schrittbausteines:

- übersichtlicher Code,
- mögliche Schritte sind 0..999.

Der Quellcode wird für ein leichtes Editieren ausführlich mit Kommentaren versehen.



## 4.6. Quellengenerator- AWL- Schrittkette- Extended

The screenshot shows a dialog box titled "AWL-Schrittkette-Extended". It has several input fields and checkboxes. The fields are: FC-Nummer (80), FC-Namen (FC\_Schrittkette), DB-Nummer (80), DB-Namen (DB\_Schrittkette), FC-Nummer (Timer) (81), FC-Namen (Timer) (FC\_SchrittTimer), and a dropdown for "Max. Anzahl Schritte (viele Schritte=> großer DB)" set to 64. There are also fields for "Nr. 1. Schritt" (0), "Nr. Initialschritt" (0), "Anzahl Schritte" (12), and "Schrittbestand" (5). Checkboxes for "Erzeuge interne Timer" and "Erzeuge Rückwärts-Transitionen" are present. At the bottom, there are four "In-Variable" fields: "Freigabe 1" (Auto), "Freigabe 2" (Freigabe), "Freigabe 3 (vorw.)" (SV), and "Freigabe 4 (rückw.)" (SR). Buttons for "Beispiel 1", "Beispiel 2", "Abbrechen", and "Generieren" are on the right.

Hiermit kann das Gerüst für eine Schrittkette erzeugt werden. Benötigte Symbole werden parallel dazu erstellt.

Einige Eigenschaften des erzeugten Schrittbausteines:

- es können mehrere Schritte zu gleich aktiv sein,
- mehrere Initialschritte sind erlaubt,
- alternativ- und Simultanverzweigungen sind möglich,
- Rückwärtsschritte sind möglich,
- jeder Schritt kann mit einem Timer (D oder L) versehen werden,
- geringer Speicherverbrauch.

Der Quellcode wird für ein leichtes Editieren ausführlich mit Kommentaren versehen.

## 4.7. Adressrechner

Schnell ermitteln, wie viele Bits zwischen zwei Adressen liegen. Oder die nächste Adresse nach n Bits ermitteln. Das geht mit dem Adressrechner.

In der Eingabemaske gibt es 4 Werte. Der durch X gekennzeichnete ist der gesuchte, und wird automatisch berechnet.

Durch einen Klick auf den Button vor einem Eingabefeld, kann dieses zum „gesucht- Feld“ gemacht werden.

## 4.8. Pointer- Tool

Hier lassen sich Pointer und ANY- Pointer mit wenigen Mausklicks, in gewünschter AWL-Schreibweise (Byte, Word, DWord) zusammensetzen. Die einzelnen Programmzeilen können automatisch kommentiert werden.



## 5. Lizenzrechte

**Die Versionen 3.XX sind FREeware :-)  
Es ist kein Lizenzschlüssel erforderlich!**

Tool4S7 ist ein im Sinne des deutschen Urheberrechtes geschütztes Programm. Sie dürfen es jedoch unter bestimmten Bedingungen weitergeben. Die folgenden Einschränkungen sind zu beachten:

- Das Programm darf nur in unveränderter Form und nur unter dem Namen Tool4S7 weitergeben werden. Eine Umbenennung ist nicht zulässig.
- Die Weitergabe des Registrierungsschlüssels ist VERBOTEN (Datei: Tool4S7.key).
- Das Programm darf NICHT verkauft werden. Ein Unkostenbeitrag für Datenträger bleibt davon unberührt.
- Das Programm darf nur als Paket mit allen dazu gehörigen Dateien (außer Registrierungsschlüssel) weitergegeben werden.
- Verändern und Rückverfolgen des Programmcodes ist nicht gestattet.
- Alle in diesem Programm genannten Markenzeichen sind Eigentum des jeweiligen Besitzers.

### FUNKTIONSUMFANG

Mit der unregistrierten Version stehen dem Benutzer alle Funktionen des Programmes zur Verfügung, um diese kennen zu lernen und zu testen. Lediglich die Anzahl der erzeugbaren Zeilen ist teilweise begrenzt. Durch Aktivierung des Registrierungsschlüssels wird diese Einschränkung entsprechend dem "Schlüsselstatus" dauerhaft beseitigt.

Sollten Erweiterungen notwendig sein und diese in das Konzept von Tool4S7 passen, so ist der Autor bestrebt, diese nachträglich einzubauen.

### HAFTUNG FÜR SCHÄDEN

Der Anwender ist verpflichtet die erzeugten Datensorgfältig zu überprüfen.

Der Autor dieses Programmes lehnt jegliche Haftung für Schäden an Maschinen, verschnittenen Werkstücken und ähnlichem grundsätzlich ab.

## 6. Änderungen/ Erweiterungen

Es gibt noch viele Möglichkeiten, Tool4S7 zu erweitern und zu verbessern.

Wer Änderungswünsche, Verbesserungsvorschläge oder Fehler entdeckt, kann diese gerne senden an:

[holger.reither@web.de](mailto:holger.reither@web.de) .

Die aktuellste Version findet man im Internet unter:

<http://reither.ohost.de/tool4s7.htm> .