

ToString

Die *ToString* Funktion konvertiert eine 32-Bit-Ganzzahl in einen ASCII-String.

Funktionsprototyp:

```
char *ToString(char *buffer, long int value, char left, char length, char comma);
```

Kurzbeschreibung:

Grundsätzlich füllt die Funktion einen definierten Bereich in einem ASCII-String mit Leerzeichen auf. Anschliessend wird die Zahl rechtsbündig in den String eingefügt. An entsprechender Stelle wird zusätzlich ein ASCII-Komma eingefügt.

Rückgabewert:

Der Rückgabewert ist gleich dem Aufruf-Parameter `buffer`. Er kann also direkt weiterverwendet werden.

Parameter:

Name	Beschreibung	Werte-Bereich
<code>*buffer</code>	Zeiger auf den String-Buffer (ASCII-String)	<i>Speicher-Adresse</i>
<code>value</code>	Darzustellende 32-Bit-Ganzzahl	-2'147'483'648 ... + 2'147'483'647
<code>left</code>	Position im String, an welcher die Zahl eingefügt werden soll	0...
<code>length</code>	Anzahl ASCII-Zeichen, die die Zahl im String belegen soll	0...
<code>comma</code>	Anzahl Stellen, die nach dem Komma dargestellt werden sollen	0...

Beispiel:

In diesem Beispiel wurde eine in Millivolt gemessene Spannung auf der Standard-Ausgabe ausgegeben. Zuerst wird in einem String ein Text vorbereitet. Anschliessend wird mit Hilfe der *ToString*-Funktion die Zahl in den String integriert.

C-Code:

```
// String
// Note: Anstelle der x und des Kommas können beliebige Zeichen stehen
char buffer[13] = "U = xx,xxx V";

// Auszugebender Wert
long int wert = 16235; // 16,235 V ( = 16235mV)

// In String wandeln und ausgeben
printf(ToString(buffer, wert, 4, 6, 3));

// Auf der Standardausgabe wird folgender Text ausgegeben:
// U = 16,235 V
```

Die letzte Zeile könnte auch auf zwei Zeilen implementiert werden:

```
// In String wandeln
ToString(buffer, wert, 4, 6, 3);

// Den String nun ausgeben
printf(buffer);
```