

## Herstellen von A-Wörtern

Bisher haben wir uns nur darum gekümmert, wie man F-Wörter erzeugt. Dies geschah über die Doppelpunktdefinition im Rahmen des üblichen Interpretier- und Kompilervorgangs. Für viele Anwendungen ist das auch ausreichend. Es kann aber vorkommen, dass die im Vokabular zur Verfügung gestellten Wörter nicht ausreichen. In diesem Fall ist es zweckmäßig, das Problem genau zu lokalisieren und dafür ein passendes A-Wort selbst zu erzeugen. Wir wollen dies an einem einfachen Beispiel verdeutlichen.

Die Aufgabe möge darin bestehen, ein Wort zu erzeugen, welches sämtliche Werte auf dem Stack über die serielle Schnittstelle ausgibt; ein solches Wort könnte gut zu Testzwecken eingesetzt werden. Schleifenstrukturen und Wörter für die COM-Ausgabe und das Arbeiten mit dem SRAM stehen im Vokabular schon zur Verfügung. Was noch fehlt, ist ein Wort, das die Anzahl der Werte auf dem Stack angibt; diese Anzahl bezeichnet man manchmal auch als **Stacktiefe**.

Zur Ermittlung dieser Stacktiefe soll nun ein neues Wort `stackcount` erzeugt werden. Hierzu muss auf den Stackpointer `X` zurückgegriffen werden; da der bislang durch kein Forth-Wort erfasst ist, muss `stackcount` mit Maschinencode erzeugt werden. Wir müssen also ein A-Wort herstellen.

Der nötige Assemblercode sieht so aus:

```
.def stackcount = r16      ; r0 bis r15 reserviert für  
                          ;                               Interrupts  
mov stackcount, XL        ; Stackpointer XL nach stackcount  
subi stackcount, $60      ; Stack-Startadresse $60 subtrahieren  
st x+, stackcount        ; Ergebnis auf Stack legen  
ret
```

Wir assemblieren den Code, zum Beispiel mit Studio 4, und speichern die Hex-Datei unter dem Namen "stackcount.hex" ab. Nun öffnen wir unser Programm MikroForth und klicken auf die Lasche "Vokabular editieren". Auf der Navigationsleiste klicken wir die +-Schaltfläche an; es öffnet sich ein Fenster zum Editieren von Wörtern; dort tragen wir den Namen des Wortes und den Kommentar so ein wie in Abb. 1 zu sehen.

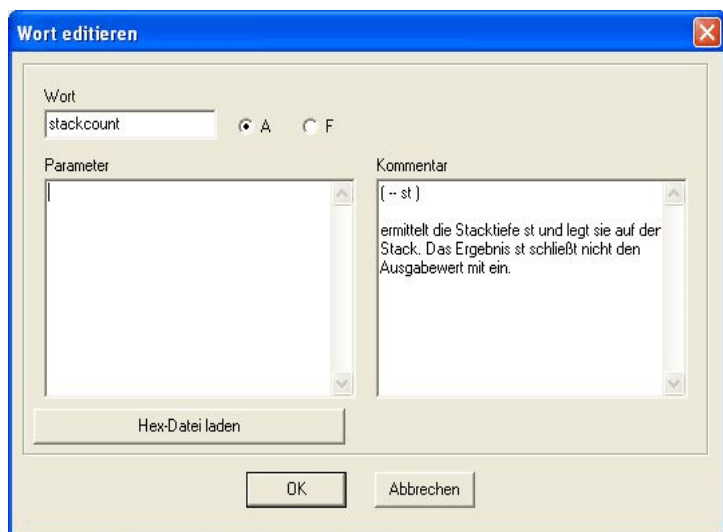
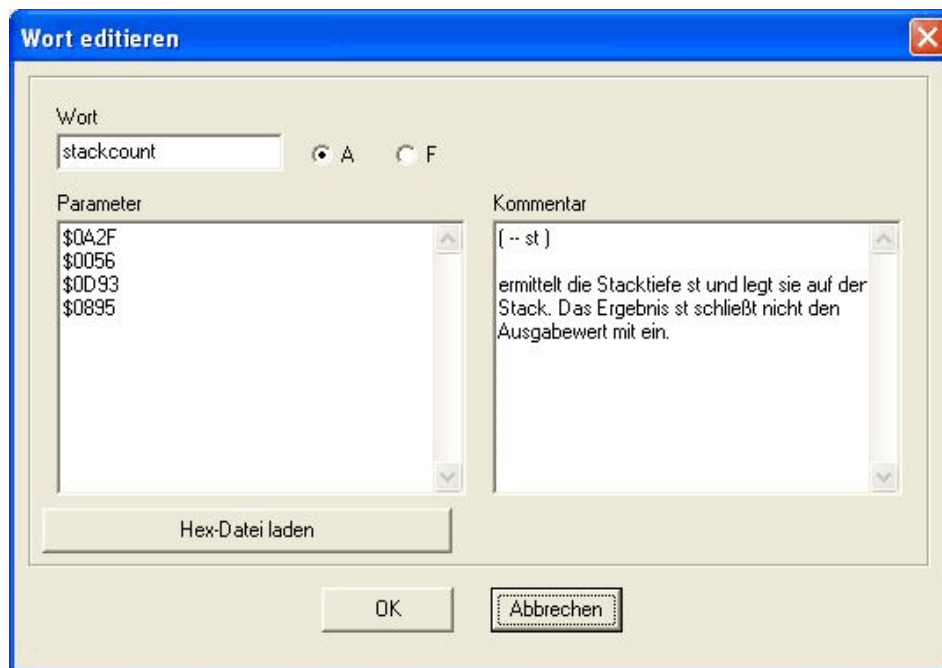


Abbildung 1

Nun laden wir den vorbereiteten Hex-Code in das Parameterfeld: Dazu betätigen wir die Schaltfläche “Hex-Datei laden” unterhalb des Parameterfeldes und öffnen unsere Datei “stackcount.hex”. Das Editier-Fenster sieht dann so aus:



**Abbildung 2**

Jetzt bestätigen wir noch die Eingabe mit der OK-Schaltfläche. Damit ist unser neues A-Wort fertig. Es befindet sich allerdings nur im temporären Vokabular. Um es dauerhaft zu speichern, klicken wir abschließend auf die Disketten-Schaltfläche in der Navigationsleiste.

Mithilfe von `stackcount` können wir jetzt leicht das Wort `stack2com` schreiben, welches den Stack über die COM-Schnittstelle ausgibt:

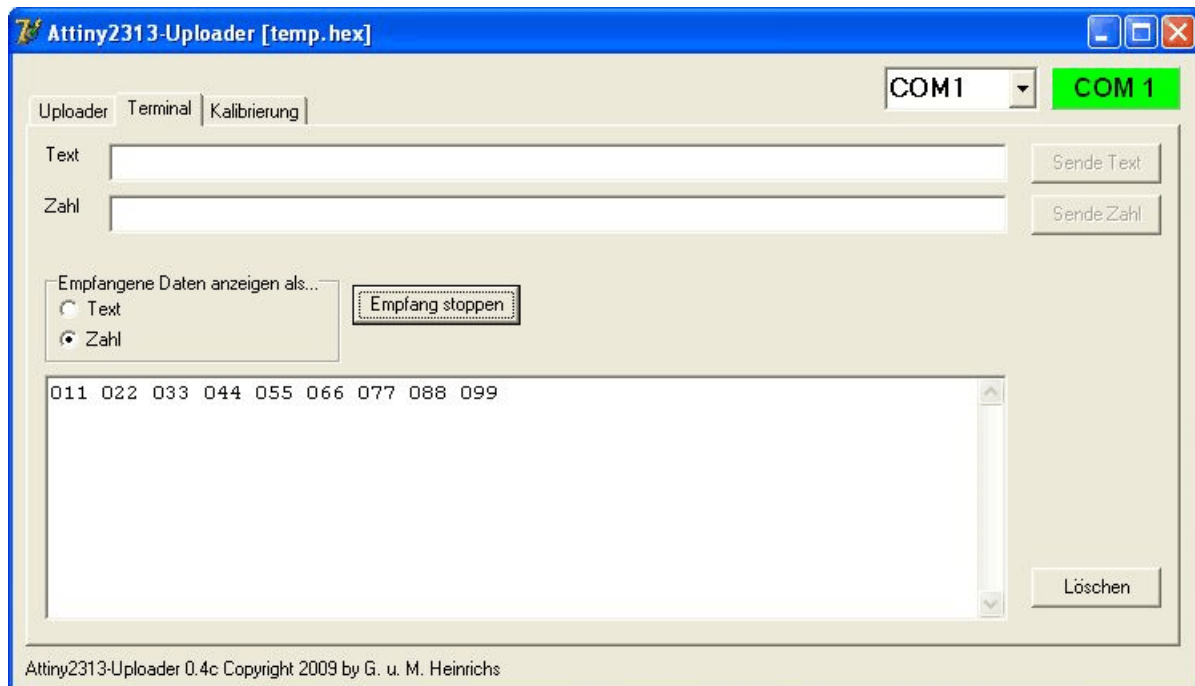
```
: stack2com initcom stackcount 1 do I 95 + sram> >com 250  
                                waitms loop ;
```

Was geschieht hier? Zunächst wird die COM-Schnittstelle initialisiert; anschließend wird die Stacktiefe ermittelt und auf den Stack gelegt. Zusammen mit der folgenden Zahl 1 bildet sie End- und Startwert der folgenden `do-loop`-Schleife. Innerhalb der Schleife wird zum Schleifenindex jeweils die Zahl 95 addiert; das Ergebnis ist die Adresse des jeweiligen Stackregisters im SRAM. Der Inhalt dieses Registers wird dann mit dem Befehl `sram>` auf den Stack gelegt und von `>com` an die serielle Schnittstelle weitergereicht. Eine kurze Wartezeit - wenn auch nicht so lange, wie hier angegeben - ist erforderlich, damit die COM-Übertragung eines Bytes nicht durch die Übertragung des nächsten gestört wird.

Zum Austesten benutzen wir das folgende `main`-Wort:

```
: main 11 22 33 44 55 66 77 88 99 stack2com ;
```

Hierdurch werden die Zahlen 11, 22 bis 99 auf den Stack gelegt und gleich darauf durch `stack2com` ausgegeben. Das können wir leicht nachkontrollieren, indem wir im Uploader-Programm das Terminal-Programm aktivieren.



**Abbildung 3**