

Rekursion mit MikroForth

Das einfache Konzept unseres FORTH-Compilers bedingt, dass ein Forth-Wort sich nicht direkt selbst aufrufen kann. Lassen wir z. B. das folgende Programm

```
: rektest 1 . rektest ;  
: main rektest ;
```

interpretieren, so erhält man die Meldung:

```
Fehler: Das Wort "rektest" wurde im Vokabular nicht gefunden.  
"rektest" wurde nicht im Vokabular eingetragen!  
Warnung: Interpretiervorgang abgebrochen...
```

Allerdings können wir die Kenntnisse über die Art und Weise, wie MikroForth arbeitet, ausnutzen, um mit einem Trick doch noch eine Rekursion zu realisieren. Dazu rufen wir aus der Doppelpunktdefinition von `rektest` nicht das Wort `rektest` selbst auf, sondern ein anderes Wort `zu_rektest`, welches seinerseits dafür sorgt, dass `rektest` ausgeführt wird. Allerdings kann das Wort `zu_rektest` das Wort `rektest` nicht wie üblich aufrufen; dies hätte nur eine Fehlermeldung wie oben zur Folge.

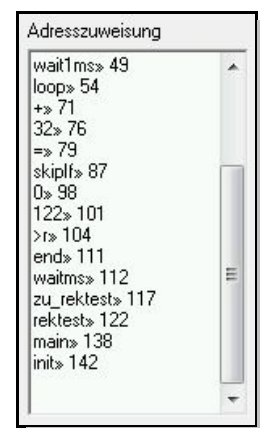
Wie kann dieser Aufruf von `rektest` nun anders realisiert werden? Hierzu nutzen wir aus, dass die Adresse des Wortes, das als nächstes auszuführen ist, auf dem Returnstack zu liegen hat. Das Wort `zu_rektest` muss also nichts anderes leisten, als die Adresse von `rektest` auf den Returnstack zu legen.

An einem konkreten Beispiel soll die Vorgehensweise erläutert werden. Die Zahlen von 1 bis 32 sollen ausgegeben werden, zuerst vorwärts zählend und dann rückwärts.

```
: zu_rektest 0 122 >r >r ;  
: rektest dup dup . 255 waitms 1 + dup 32 equal skipIf  
                                          zu_rektest . 255 waitms ;  
: main 1 rektest . ;
```

Das Wort `zu_rektest` legt die Adresse von `rektest` auf den Returnstack. Diese Adresse besteht hier aus dem Highbyte 0 und dem Lowbyte 122. Wie gelangt man an diese Adresse? Dazu braucht man nach dem Kompilieren nur die Adresszuweisungstabelle anschauen. Die drittletzte Zeile in Abb. 1 zeigt, dass die gesuchte Adresse 122 ist. Für die Adressfindung ist es nicht gleichgültig, welche Werte für die beiden Adressbytes in der Definition von `zu_rektest` zunächst eingesetzt worden sind. Eine nachträgliche Kontrolle mit der gefundenen Adresse empfiehlt sich also.

Die Rekursionstiefe wird durch die Größe des SRAMs begrenzt.



Word	Address
wait1ms»	49
loop»	54
+»	71
32»	76
=»	79
skipIf»	87
0»	98
122»	101
>r»	104
end»	111
waitms»	112
zu_rektest»	117
rektest»	122
main»	138
init»	142

Abbildung 1