

## MikroForth-Variablen

Zum kurzfristigen Speichern und zur Übergabe von Werten von einem Wort zum anderen ist der Stack bestens geeignet. Zum langfristigen Speichern bietet sich eher das EEPROM des Attiny an. Allerdings ist es recht mühselig, sich die unterschiedlichen Adressen für die zu speichernden Werte zu merken. Abhilfe schaffen hier **Variable**. Sie werden in MikroForth folgendermaßen deklariert:

```
variable <Variablenname>
```

Die Variablendeklaration kann nicht innerhalb einer Doppelpunktdefinition stehen; das Schlüsselwort `variable` muss am Anfang einer Zeile stehen. Pro Zeile kann jeweils immer nur eine einzige Variable deklariert werden und am Ende der Zeile braucht kein Semikolon stehen.

Der Compiler vergibt für jede Variable eine Nummer zwischen 0 und 126, die als EEPROM-Adresse dient; die EEPROM-Zelle mit der Adresse 127 ist für den OSCCAL-Wert reserviert. Zudem erzeugt der Compiler für jede Variable ein Wort mit gleichem Namen. Dieses Wort hat nur eine einzige Aufgabe: Es legt die zugehörige Adresse auf den Stack.

An einem Beispiel wollen wir uns die Benutzung von Variablen anschauen:

```
variable KontoNr
: ablegen 129 KontoNr >eprom ;
: holen KontoNr eprom> ;
: main ablegen holen . ;
```

In der ersten Zeile wird die Variable `KontoNr` deklariert. In der zweiten Zeile wird durch das Wort `KontoNr` die zugehörige Adresse auf den Stack gelegt. Wenn `KontoNr` als erste Variable deklariert worden ist, dann ist diese Adresse \$00. Durch das nachfolgende Wort `>eprom` wird die Zahl 129 unter der EEPROM-Adresse \$00 gespeichert. In der dritten Zeile wird die Zahl 129 wieder aus der EEPROM-Zelle \$00 geholt und auf den Stack gelegt. Das Wort `KontoNr` kann hier wieder als Stellvertreter für die zugehörige EEPROM-Adresse angesehen werden. Durch die Einführung von Variablen ändert sich der Quellcode im Wesentlichen nicht; er wird aber viel übersichtlicher. Man beachte: Da bei einer EEPROM-Zelle nur eine beschränkte Anzahl von Schreibvorgängen durchgeführt werden kann, sollte man das Wort `holen` sparsam einsetzen. Von daher sollten die MikroForth-Variablen eher als Konstanten-Speicher angesehen werden.

Im Prinzip könnten die Variablen auch zur Adressierung von SRAM- oder auch Flash-Registern benutzt werden. Davon ist aber dringend abzuraten. Der Compiler vergibt nämlich die Adressen der Reihe nach, bei 0 beginnend. So ist es fast sicher, dass unkontrolliert wichtige Statusregister oder auch Teile des Programms auf diese Weise überschrieben würden. Die Folgen wären fatal.

*Aufgabe 1*

Geben Sie das obige Beispiel ein und testen Sie es. Schauen Sie sich mit dem Vokabular-Editor auch das Wort `KONTONr` an.