

# 13 Assemblerprogramme für Controller-Grundfunktionen

Die folgenden Programme sind mit einem Makroassembler in Maschinensprache übersetzt und in einem Hardware-Entwicklungssystem getestet worden. Der Speicherbereich für die Anwendungsprogramme liegt im Entwicklungssystem ab Adresse 8000h. Das sind die oberen 32 K des Adressbereiches.

## 13.1 Blink- und Lauflicht mit Unterprogramm-Aufruf

### Blinklicht 1

Alle Bits von Port 5 sollen mit einer sichtbaren Frequenz blinken

Assemblerprogramm:

```
;***** Blinklicht 1, Blink1.asm *****
;
;synchrone Blinken aller Ausgänge Port 5 mit sichtbarer Frequenz
;
;***** controller 80535
        mov p5,#0ffh          ;alle Ausgänge P5 ein
b11: mov r2,#0ffh          ;Wartezeit: 2 Register als 16-Bit-Zähler
t2:  mov r1,#0ffh          ;herunterzählen
t1:  djnz r1,t1
        djnz r2,t2
        mov a,p5
        cpl a
        mov p5,a
        ajmp b11
```

### Blinklicht 2

Die blinkenden Bits an Port 5 sollen mit einem Schalter an Port1.0 angehalten werden.

Assemblerprogramm:

```
;***** Blinklicht 2, Blink2.asm *****
;
;Synchrone Blinken aller Ausgänge an Port 5 mit sichtbarer Frequenz,
;wenn Einschalter P1.0 = 1.
;
;***** controller 80535
        org 8000h
        mov p5,#0ffh          ;alle Ausgänge P5 ein
b11: jnb p1.0,b11          ;Einschalter = 0? Ja: Warteschleife
```

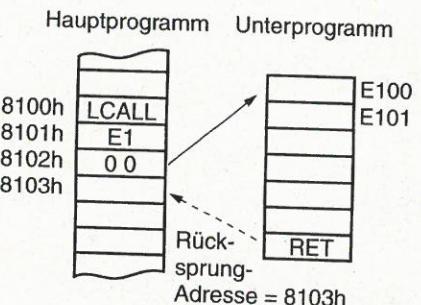
```
        mov r2,#0ffh
t2:  mov r1,#0ffh
t1:  djnz r1,t1
        djnz r2,t2
        mov a,p5
        cpl a
        mov p5,a
        ajmp b11
```

;Wartezeit: 2 Register als 16-Bit-Zähler  
herunterzählen  
;P5 invertieren

### 13.1.1 Unterprogramme

Unterprogramme werden vom Hauptprogramm mit dem Befehl ACALL oder LCALL aufgerufen. Bei der Ausführung des Befehls wird erst die Rücksprungadresse auf dem Stack gerettet. Dann gibt der Controller die Anfangsadresse des Unterprogramms aus, um dieses zu bearbeiten.

Das Unterprogramm endet mit dem Befehl RET. Dieser bewirkt ein Rückholen der Adresse vom Stack. Diese Adresse wird als nächstes ausgegeben und damit das Hauptprogramm fortgesetzt.



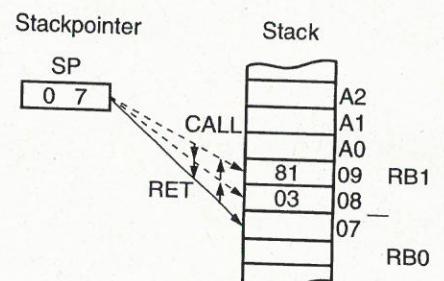
### Retten der Rücksprungadresse auf dem Stack

Der Stack befindet sich im internen RAM-Speicher. Er wird durch den Stackpointer verwaltet.

Bei einem CALL-Befehl wandert der Stackpointer zwei Adressen aufwärts. Bei einem RET-Befehl zwei Adressen abwärts. Die zu rettende Rücksprungadresse ins Hauptprogramm wird zu zweimal 8 Bit auf die Adressen im Stack gelegt, auf die der Stackpointer zeigt. Am Ende des Unterprogramms wird die Adresse durch RET wieder in den Programm-Counter zurückgeholt.

Nach einem Reset steht im Stackpointer die Adresse 07. Der Stack beginnt dann ab Adresse 08.

Der Stackpointer lässt sich durch Laden einer Konstanten auf jede Adresse innerhalb des internen RAMs legen.



## Retten von Registerinhalten auf dem Stack

Genau wie die Rücksprungadresse lassen sich auch andere Speicherinhalte des direkt adressierbaren Speicherbereiches auf dem Stack retten und nach der Bearbeitung des Unterprogramms wieder zurückholen. Mit dem PUSH-Befehl werden Inhalte auf dem Stack gerettet, mit dem POP-Befehl wieder zurückgeholt.

Retten von Speicherinhalten:  
PUSH direkt  
Funktion wie bei Befehl CALL.

Rückholen der Speicherinhalte:  
POP direkt  
Funktion wie bei Befehl RET.

## Lauflicht 1 mit Unterprogramm für Zeitwerk

```
;***** Lauflicht 1, LAUF1.ASM *****
;
;Lauflicht an Port 5
;
;*****
Controller 80535
org 8000h
setb p5.1          ;Lauflicht Ausgangsmuster
anf:lcall zeit2    ;Wartezeit
    mov a,p5          ;Lauflicht 1 Stelle links schieben
    rl a
    mov p5,a
    ljmp anf
;----- Unterprogramm Zeit2 -----
zeit2:mov r2,0ffh
    t2:mov r1,0ffh
    t1:djnz r1,t1
    djnz r2,t2
    ret
;*****
```

## Übung 13.1

Schreiben Sie ein Programm für einen umschaltbaren Frequenzgenerator. Die Frequenzen sollen im sichtbaren Bereich liegen und im Verhältnis 2 : 1 umschaltbar sein.

