Nun wird aber nicht während der Dauer der Tastenbetätigung ständig der Scancode gesendet. Die Tastatur sendet hingegen einen so genannten *Make*- und einen *Break*-Code. Der *Make*-Code wird beim Betätigen der Taste ausgelöst und der *Break*-Code beim Loslassen der Taste zum PC gesendet. *Make*- und *Break*-Code sind gleich, nur wird vor dem *Break*-Code ein hexadezimales F0 gesendet. Komplizierter ist es bei einigen speziellen Funktionstasten (z. B. PRNT SCRN, deutsche Tastenbeschriftung meist Druck). Dort wird eine Codefolge gesendet, die mit E0 beginnt.

Mit dieser kleinen Einführung sind wir in der Lage, einen Code ohne Verwendung der Tastatur zu senden. Nehmen wir an, dass ein p gesendet werden soll, so senden wir für den Tastendruck hexadezimal 4D und für das Loslassen nach einer kleinen Wartezeit F0 und 4D zum PC. Komplizierter wird es schon beim großen P. Zuerst muss die Shift- und anschließend die P-Taste gedrückt werden.

Das Loslassen der Tasten geschieht in umgekehrter Reihenfolge. Wir senden somit als *Make*-Code 12 für die linke *Shift*-Taste und anschließend ebenfalls 4D. Dann kommen die zugehörigen vier Bytes des *Break*-Codes: F0 4D F0 12. Dieses Prinzip funktioniert auch, wenn mehr als zwei Tasten betätigt werden sollen.

Der Compiler BASCOM-AVR bietet die Möglichkeit, eine PC-Tastatur zu emulieren.

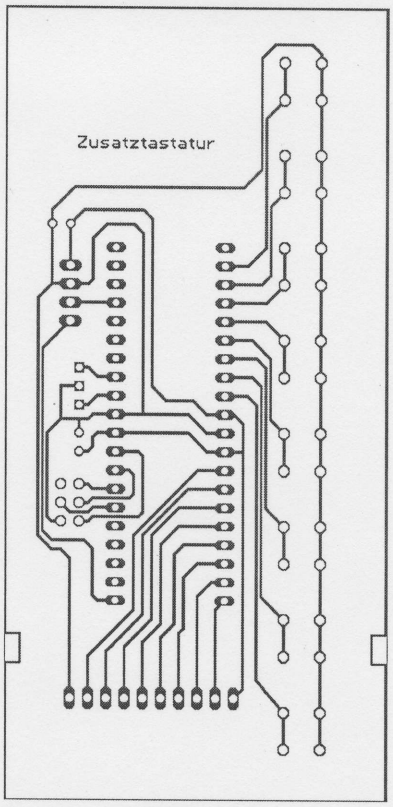

Bild 3: Layout der Tastaturplatine

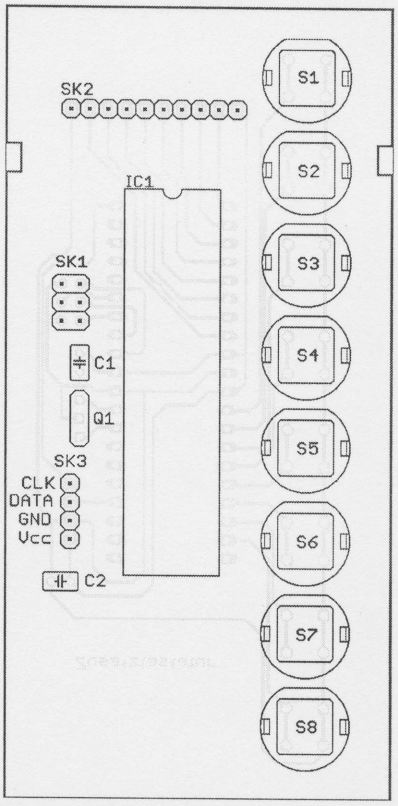

Bild 4: Bestückung der Tastaturplatine

Das heißt, ein Controller mit entsprechender Software simuliert den Tastendruck.

Tastaturemulation mit AVR

Zur Realisierung einer Zusatztastatur benötigen wir also nur einen AVR-Controller mit einigen wenigen Tasten und einem Interface in Richtung PC. Die Schaltung weist keine Besonderheiten auf. Wir verwenden einen ATmega16-Controller, um ausreichend Portleitungen für eventuelle Erweiterungen zur Verfügung zu haben. An Port C sind in der Grundversion acht Tasten angeschlossen. Externe Pullup-Widerstände sind nicht er-

Programm der Zusatztastatur

```
$regfile = "m16def.dat"          ' Controllerlibrary
$crystal = 4000000                      ' Frequenz
$baud = 19200                           ' Baudrate
$hwstack = 32
$swstack = 10
$framesize = 40
$lib "mcsbyteint.lbx"    ' nur Bytes verarbeiten
    'Variablen und Konstanten dimensionieren
Const Keys = 8   ' acht Tasten angeschlossen
Dim I As Byte
Dim Key As Byte          ' enthält Tastennummer
Config Portc = Input  ' Konfiguriere Tastenfeld
Portc = 255                     ' Pullups aktivieren
                'Konfiguriere PS/2-AT-Interface
Enable Interrupts        ' Interrupts einschalten
Config Atemu = Int1 , Data = Pind.3 ,
                                  Clock = Pinb.0
Waitms 500                      ' optional delay
                          ' rcall _AT_KBD_INIT

Do                              ' Tastenabfrage
 For I = 1 To Keys
  Key = I
  Select Case Key
   Case 1 : Debounce Pinc.0 , 0 , M1 , Sub
   Case 2 : Debounce Pinc.1 , 0 , M2 , Sub
   Case 3 : Debounce Pinc.2 , 0 , M3 , Sub
   Case 4 : Debounce Pinc.3 , 0 , M4 , Sub
   Case 5 : Debounce Pinc.4 , 0 , M5 , Sub
   Case 6 : Debounce Pinc.5 , 0 , M6 , Sub
   Case 7 : Debounce Pinc.6 , 0 , M7 , Sub
   Case 8 : Debounce Pinc.7 , 0 , M8 , Sub
  End Select
 Next
Loop
End

M1:
Sendscankbd Text1    ' sende Code aus Data
Waitms 1000
Return

M2:
Sendscankbd Text2
Waitms 1000
Return

M3:
Sendscankbd Text3
Waitms 1000
Return

M4:
Sendscankbd Text4
Waitms 1000
Return

M5:
Sendscankbd Text5
Waitms 1000
Return

M6:
Sendscankbd Text6
Waitms 1000
Return

M7:
Sendscankbd Text7
Waitms 1000
Return

M8:
Sendscankbd Text8
Waitms 1000
Return

' Datazeilen, je Taste eine Zeile beginnend
' mit Zahl der zu sendenden Zeichen

' CTRL-C , Kopieren
Text1:
Data 6 , &H14 , &H21 , &HF0 , &H21 ,
                                &HF0 , &H14

' CTRL-V , Einfügen
Text2:
Data 6 , &H14 , &H2A , &HF0 , &H2A ,
                                &HF0 , &H14

' text
Text3:
Data 12 , &H2C , &HF0 , &H2C , &H24 ,
&HF0 , &H24 , &H22 , &HF0 , &H22 ,
                              &H2C , &HF0 , &H2C

' a
Text4:
Data 3 , &H1C , &HF0 , &H1C

' A
Text5:
Data 6 , &H12 , &H1C , &HF0 , &H1C ,
                                &HF0 , &H12

' Taste F2
Text6:
Data 3 , &H06 , &HF0 , &H06

' Taste F7
Text7:
Data 3 , &H83 , &HF0 , &H83

' Einfügetaste
Text8:
Data 5 , &HE0 , &H70 , &HE0 , &HF0 , &H70
```