

# Praktikum Mikrocomputertechnik

## Versuch 3:

Kommunikation mit MF II Tastatur

Betreuer:

Armin Schön

Termin:

\_\_\_\_\_

Teilnehmer:

Gruppe:

\_\_\_\_\_

Semester:

\_\_\_\_\_

Student 1:

\_\_\_\_\_

Student 2:

\_\_\_\_\_

Student 3:

\_\_\_\_\_

Testat:

Datum:

\_\_\_\_\_

Unterschrift:

\_\_\_\_\_

Bemerkungen:

## 1.1 Kommunikation mit MF II-Tastatur

Dieser Abschnitt erläutert die Kommunikation mit einer MF II-Tastatur über den Mikrocontroller SAB-C167CR-LM.

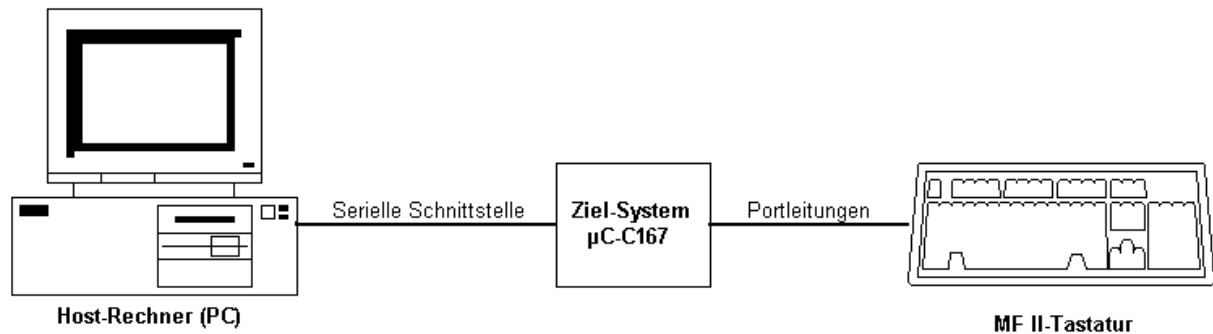


Abb.1. Prinzipieller Versuchsaufbau

### 1.1.1 Versuchsinhalte

- Versuch Nr. 1: Aufzeichnung des Protokolls der Tastatur nach C167-Übertragung mit einem Digitalspeicheroszilloskop (DSO)
- Versuch Nr. 2: Aufzeichnung der Make- und Break-Codes einer MF II-Tastatur mit dem Digitalspeicheroszilloskop
- Versuch Nr. 3: Ziffern 0...9 im Polling-Betrieb von der Tastatur einlesen
- Versuch Nr. 4: Ziffern 0...9 im Interrupt-Betrieb von der Tastatur einlesen
- Versuch Nr. 5: Senden von Befehlen an die MF II-Tastatur

### 1.1.2 Grundlagen zur MF II-Tastatur

In diesem Abschnitt sollen grundlegende Kenntnisse zur MF II-Tastatur vermittelt werden, die die Studenten zum Durchführen des Praktikumsversuches benötigen.

### 1.1.2.1 Allgemeines

Die heute allgemein benutzten MF II-Tastaturen (Multifunktionstastaturen) sind komplexere Geräte, als dies von außen zunächst den Anschein hat. Dies zeigt sich schon daran, dass jede Tastatur einen Tastaturchip (Abb. 3) besitzt. Dieser Chip überwacht die sog. Scanmatrix (Abb. 2), die hauptsächlich aus sich kreuzenden Leitungen besteht, an deren Kreuzungspunkten ein kleiner Schalter (Kontakt) sitzt. Beim Betätigen einer Taste wird dieser Schalter geschlossen.

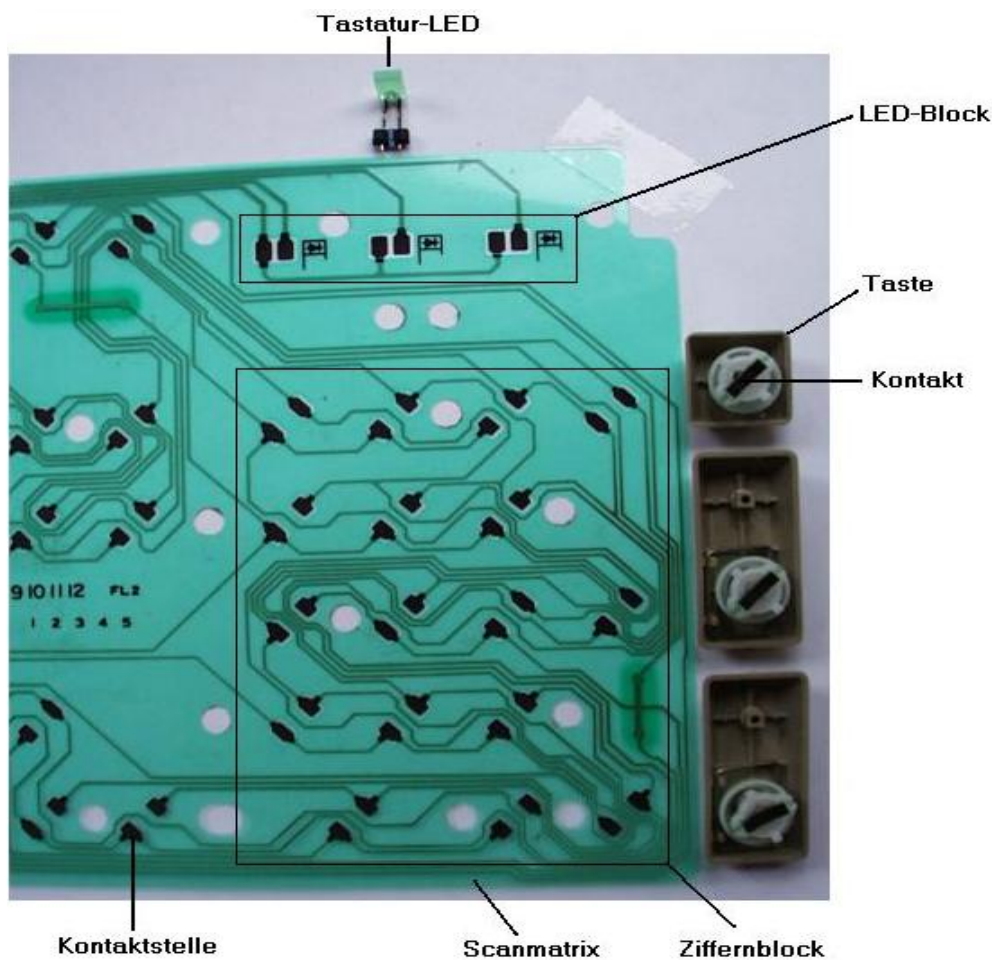
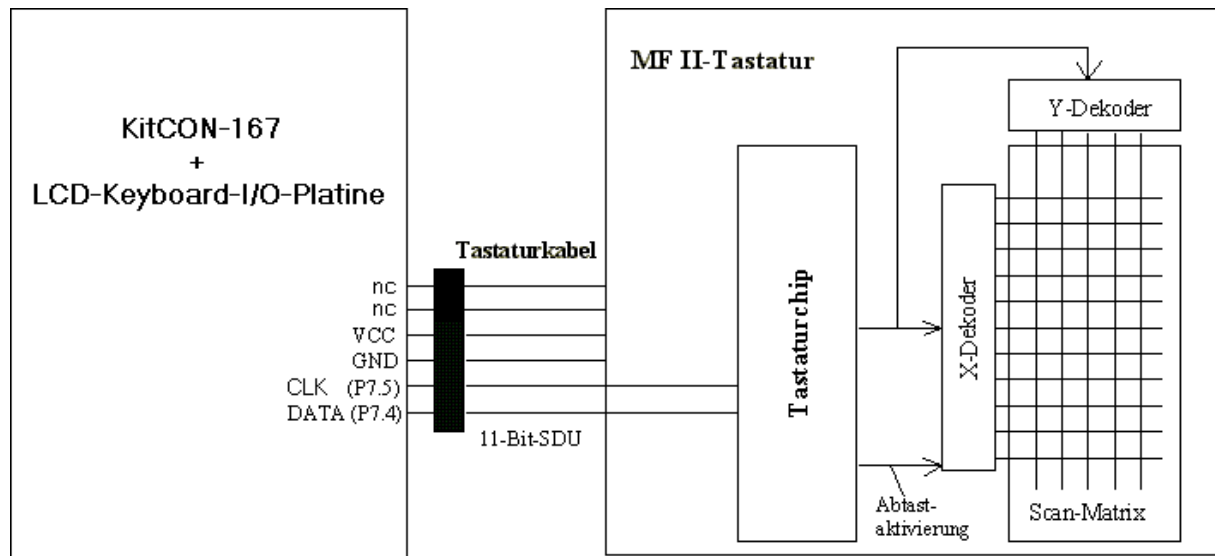


Abb. 2. Scanmatrix (Ziffernblock)

Das im Tastaturchip ablaufende Mikroprogramm ist in der Lage, ein schnell ablaufendes Prellen einer Taste von einer gewollten und langsamer ablaufenden mehrfachen Tastenbetätigung durch den Benutzer zu unterscheiden.

In Abb. 3 ist die Verbindung des Mikrocontrollers mit der MF II-Tastatur dargestellt.

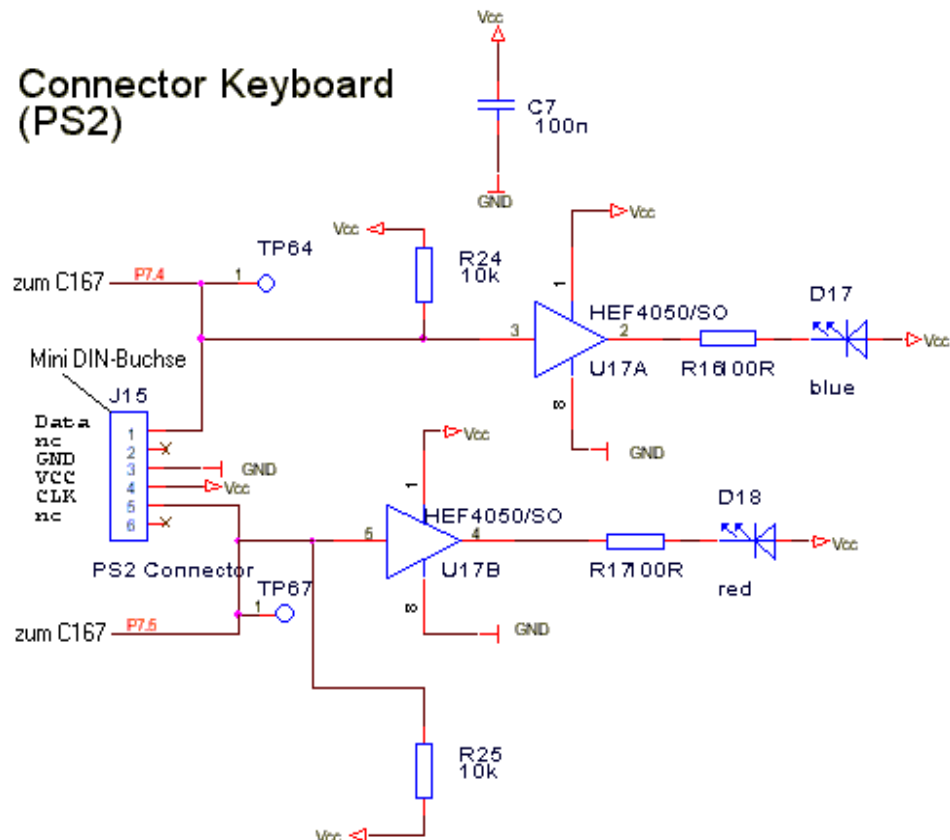


**Abb. 3.** Verbindung des Mikrocontrollers SAB-C167CR-LM mit der MF II-Tastatur

Der Tastaturchip muss in regelmäßigen Abständen den Zustand der Scanmatrix überprüfen, um den Zustand *open* oder *geschlossen* der Schalter zu ermitteln. Dies wird dadurch erreicht, dass er nacheinander einzeln die X-Leitungen aktiviert und erfasst, von welchem Y-Anschluss er ein Signal empfängt. Durch diese X- und Y-Koordinaten ist die neu gedrückte oder losgelassene Taste eindeutig identifizierbar. Der Tastaturchip stellt nun fest, welche Taste gedrückt oder losgelassen wurde und legt einen entsprechenden Code in einem tastaturinternen Puffer ab. Im Anschluss daran überträgt die Tastatur den Code als seriellen Datenstrom (Abschnitt 1.1.2.4) über das Tastaturkabel an den PC bzw. hier im Versuch zum Mikrocontrollerboard.

### 1.1.2.2 Anschluss der Mini-DIN-Tastaturbuchse auf der LCD-Keyboard-I/O-Platine

In der Abb.4 ist die elektrische Verbindung der Mini-DIN-Tastaturbuchse mit dem Mikrocontroller SAB-C167CR-LM zu sehen.



**Abb.4.** Anschluss der Mini DIN-Tastaturbuchse auf der LCD-Keyboard-I/O-Platine

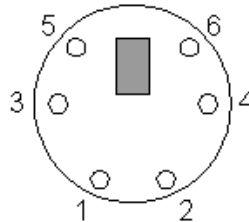
HEF4050 dienen als Treiber für LEDs D17 und D18

TP64: Testpunkt für Serielle Daten (DATA)

TP67: Testpunkt für Tastaturtakt (CLK)

### 1.1.2.3 Mini DIN-Tastaturstecker

Abb. 5 zeigt den zugehörigen Mini DIN-Tastaturstecker [15].



**Abb. 5.** Mini DIN-Tastaturstecker (Sicht in den Stecker)

In der nachfolgenden Tabelle 1 ist die Pinbelegung des Tastatursteckers (Mini DIN) dargestellt.

<b><i>Pin</i></b>	<b><i>Signal</i></b>
1	Tastaturdaten (DATA)
2	Frei (nc = not connected)
3	Signalmasse (GND)
4	VCC (+5V)
5	Tastaturtakt (CLK)
6	Frei (nc = not connected)
Abschirmung (Shield)	Gehäusemasse (shell)

**Tabelle 1.** Pinbelegung des Tastatursteckers (Mini DIN)

### 1.1.2.4 Übertragungsprotokoll Tastatur -> C167

Das 11-Bit-SDU (**S**erial **D**ata **U**nit, dt. serielle Dateneinheit) wird auf der DATA-Leitung (P7.4 des C167) übertragen. Die CLK-Leitung (P7.5 des C167) liefert ein Taktsignal für den Datenaustausch mit dem PC bzw. Mikrocontroller, so dass das 11-Bit-SDU synchron übertragen werden kann.

In Abb. 6 ist der Aufbau des 11-Bit-SDU zu sehen.

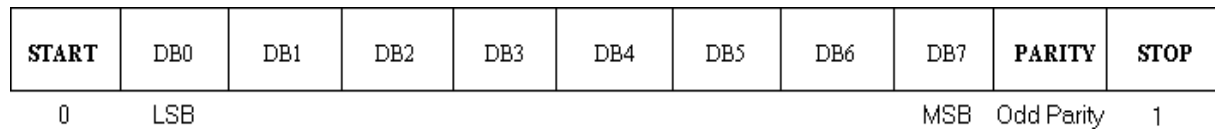


Abb. 6. 11-Bit-SDU

- START : Startbit (stets gleich 0)
- DB0...DB7 : Datenbit 0 bis 7  
             DB0 = LSB (**L**east **S**ignificant **B**it, dt. niederwertigstes Bit)  
             DB7 = MSB (**M**ost **S**ignificant **B**it, dt. höchstwertigstes Bit)
- PARITY : Odd Parity (ungerade Parität)
- STOP : Stopbit (stets gleich 1)

Abb. 7 zeigt die zeitliche Abfolge der einzelnen Bits des Tastatur-SDUs.

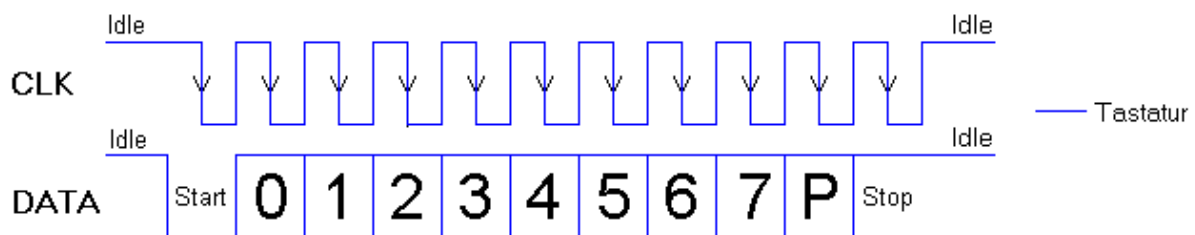


Abb. 7. Übertragungsprotokoll Tastatur -> C167

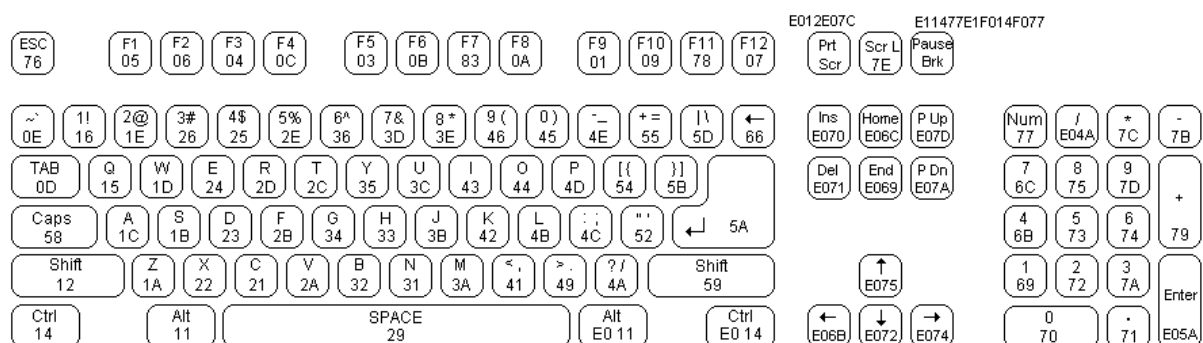
Wie in Abb. 7 zu erkennen ist, erzeugt der Tastaturchip ein Taktsignal (CLK), bei dessen fallender Flanke das gerade auf der Datenleitung (DATA) anstehende Bit gültig ist und zur Übernahme durch den Mikrocontroller bereitsteht.

Als erstes wird das Startbit (Logikpegel 0) ausgesendet. Im Anschluss daran folgen nacheinander die 8 Datenbits (DB0...DB7), wobei das niederwertigste Bit (LSB) als erstes ausgegeben wird. Nach den Datenbits wird von der Tastatur das sogenannte Paritätsbit (P) übertragen, das dem Empfänger erlaubt festzustellen, ob während der Übertragung Fehler aufgetreten sind oder nicht. Liegt eine fehlerfreie Übertragung vor, muss die Anzahl der logischen Einsen (ohne Stopbit) stets ungerade sein (ungerade Parität). Zum Schluss wird noch das Stopbit (Logikpegel 1) ausgesendet, das das Ende der Übertragung von der Tastatur zum Mikrocontroller SAB-C167CR-LM

kennzeichnet. Im Anschluss daran nehmen die beiden Übertragungsleitungen (CLK und DATA) wieder den Idle-Zustand (Logikpegel 1) an.

### 1.1.2.5 Scancodes

Die Datenbits DB0 bis DB7 aus Abb. 6 stellen den sogenannten Scancode dar. Zur Identifizierung ist jeder Taste der Tastatur ein eindeutiger Scancode zugeordnet. Der Tastaturtreiber im PC setzt diesen empfangenen Scancode in das entsprechende Zeichen um. Dies ist jedoch nicht immer ein ASCII-Zeichen, da etwa für die Funktionstaste „F1“ kein ASCII-Code existiert. Außerdem ist es notwendig, in Abhängigkeit von den gedrückten Umschalttasten (SHIFT, STRG, usw.), verschiedene Zeichen auszugeben. In diesem Versuch wird der empfangene Scancode vom Mikrocontrollerprogramm, das hier in gewisser Weise als Tastaturtreiber fungiert, in das entsprechende Zeichen umgewandelt.



In Abb. 8 ist der im Versuch verwendete Scancodesatz dargestellt.

**Abb. 8.** Verwendeter Scancodesatz (Scancodesatz 2)

Die Scancodes in obiger Abbildung sind mit ihren Hexadezimalwerten dargestellt.

Zum besseren Verständnis des Sachverhaltes sollen im folgenden zwei Beispiele genauer erläutert werden. Zunächst wird erklärt, was passiert, wenn eine einzelne Taste gedrückt wird (Tabelle 2). Anschließend wird der Ablauf beim Drücken und Loslassen einer Tastenkombination beschrieben (Tabelle 3).



- Drücken einer Einzeltaste (hier „t“)

Durchgeführte Aktion	Gesendeter Tastaturcode	Erläuterung
Taste „t“ drücken und halten	2Ch..2Ch..2Ch..2Ch..usw.	Make-Code (Scancode) x-mal senden
Taste „t“ loslassen	F0h..2Ch	1. Break-Code (F0h) senden 2. Make-Code (2Ch) noch einmal senden

**Tabelle 2.** Ablauf beim Drücken einer Einzeltaste

Die Tastatur sendet beim Drücken und Halten einer Taste fortlaufend den sogenannten Make-Code, der dem Scancode der Taste entspricht. Wird die Taste wieder losgelassen, überträgt die Tastatur als erstes den Break-Code (F0h) und im Anschluss daran noch einmal den Make-Code der gerade losgelassenen Taste.

Dieser zum Schluss gesendete Make-Code ist vor allem beim nächsten Beispiel (Drücken einer Tastenkombination) interessant, da hierbei festgestellt werden muss, welche der gedrückten Tasten als erste losgelassen wurde.

- Drücken einer Tastenkombination (hier SHIFT + „t“)

Durchgeführte Aktion	Gesendeter Tastaturcode	Erläuterung
Linke <b>SHIFT</b> -Taste drücken und halten	12h..12h..12h..12h..usw.	Make-Code (Scancode) x-mal senden, bis zusätzliche Taste gedrückt wird
Zusätzlich Taste „t“ drücken und halten	2Ch..2Ch..2Ch..2Ch..usw.	Make-Code x-mal senden
Linke SHIFT-Taste loslassen	F0h..12h..2Ch..2Ch..usw.	1. Break-Code (F0h) senden 2. Make-Code (12h) der losgelassenen Taste noch einmal senden 3. Make-Code (2Ch) der Taste „t“ senden, da weiterhin gedrückt
Taste „t“ loslassen	F0h..2Ch	1. Break-Code (F0h) senden 2. Make-Code (2Ch) noch einmal senden

**Tabelle 3.** Ablauf beim Drücken einer Tastenkombination

### 1.1.2.6 Übertragungsprotokoll C167 -> Tastatur

In den vorangegangenen Abschnitten wurde erläutert, wie die Tastatur Make- und Break-Codes an den Mikrocontroller SAB-C167CR-LM sendet. Bei den MF II-Tastaturen ist es jedoch auch möglich, Befehle an den Tastaturchip zu schicken. Das Übertragungsprotokoll unterscheidet sich hierbei von der Tastatur nach C167-Übertragung (Abschnitt 1.1.2.4).

Die Abb. 9 zeigt das zugehörige Übertragungsprotokoll.

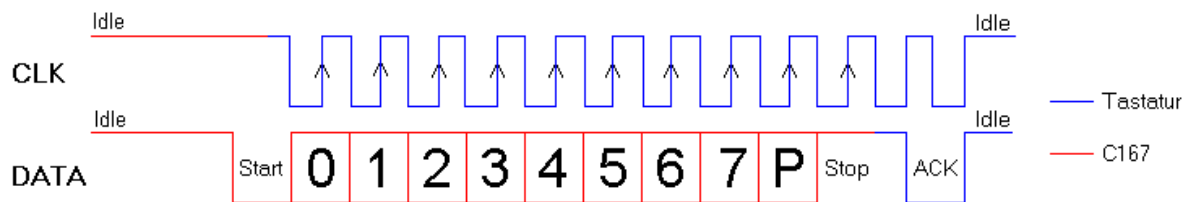


Abb. 9. Übertragungsprotokoll C167 -> Tastatur

Der Mikrocontroller SAB-C167CR-LM initiiert die Übertragung zur Tastatur, in dem er die DATA-Leitung auf Low-Pegel (Startbit) zieht. Die Tastatur erkennt dies und erzeugt daraufhin automatisch ein Taktsignal auf der CLK-Leitung. Der Mikrocontroller gibt nun bei den nächsten acht fallenden Taktflanken jeweils ein Datenbit (0...7) auf der DATA-Leitung aus. Hierbei wird das niederwertigste Bit (LSB) des Befehles als erstes gesendet. Bei der neunten und zehnten fallenden Flanke auf der CLK-Leitung gibt der SAB-C167CR-LM noch das Paritäts- und Stopbit auf die DATA-Leitung aus. Die Daten-, Paritäts- und Stopbits werden von der Tastatur mit der steigenden Taktflanke übernommen. Nachdem das Stopbit erfolgreich übertragen wurde, wird die DATA-Leitung der Tastatur überlassen, die sie daraufhin für die nächste Taktperiode auf Low-Pegel zieht. Dies entspricht dem Quittungssignal (ACK = **A**cknowledge). Zum Abschluss der Übertragung werden die beiden Leitungen (CLK und DATA) von der Tastatur auf High-Pegel gelegt, was den sog. Idle-Zustand repräsentiert.

#### 1.1.2.7 Befehle der MF II-Tastatur

In der nachfolgenden Tabelle 4 sind die Befehle [16] aufgelistet, die vom Mikrocontroller SAB-C167CR-LM aus an die MF II-Tastatur gesendet werden können. Allgemein gilt, dass die Tastatur nach jedem Befehl außer Echo und Resend ein ACK entsprechend FAh zurückgibt. Bei diesem ACK-Signal handelt es sich nicht um das Quittungssignal aus Abb. 9, sondern um einen Rückgabecode der Tastatur. Diese Rückgabecodes sind in Tabelle 5 aufgelistet.

Code	Befehl	Beschreibung
EDh	Leuchtdioden schalten	Schaltet die Leuchtdioden der MF II-Tastatur ein

		oder aus
EEh	Echo	Gibt ein Byte EEh zurück
F0h	Scancodes einstellen/ i- dentifizieren	Stellt einen von drei Scancodesätzen ein und i- dentifiziert den aktuellen Scancodesatz
F2h	Tastatur identifizieren	Identifiziert die Tastatur (ACK = AT-Tastatur, ACK+ABh+41h = MF II-Tastatur)
F3h	Wiederholungsrate /Verzögerung einstellen	Stellt die Wiederholungsrate und Verzögerung der Tastatur ein
F4h	Aktivierung	Aktiviert die Tastatur
F5h	Standard/deaktiviert	Stellt die Standardwerte ein und deaktiviert die Tastatur
F6h	Standard/aktiviert	Stellt die Standardwerte ein und aktiviert die Tas- tatur
FEh	Resend	Die Tastatur übergibt dem Tastaturchip nochmals das zuletzt übermittelte Zeichen
FFh	Reset	Führt einen internen Tastatur-Reset und an- schließend den BAT ( <b>B</b> asic <b>A</b> ssurance <b>T</b> est) aus

**Tabelle 4.** Befehle der MF II-Tastatur

#### 1.1.2.8 Indikatorbyte für Befehl „Leuchtdioden schalten“

Nachdem die MF II-Tastatur den Befehl „Leuchtdioden schalten“ übernommen hat, sendet sie einen ACK-Rückgabecode an den Mikrocontroller C167, bricht die Abtastung der Scanmatrix ab und wartet auf das Indikatorbyte, das sie ebenfalls über den Mikrocontroller C167 erhält. Dieses Indikatorbyte bestimmt die LEDs, die auf der Tastatur ein- oder ausgeschaltet werden sollen.

In Abb. 10 ist der Aufbau des Indikatorbytes [16] zu sehen.

7	6	5	4	3	2	1	0
0	0	0	0	0	CPSL	NUML	SCRL

**Abb. 10.** Indikatorbyte für Befehl „Leuchtdioden schalten“ (EDh)

CPSL : LED für CapsLock oder ShiftLock (1 = ein, 0 = aus)

NUML : LED für NumLock (1 = ein, 0 = aus)

SCRL : LED für ScrollLock (1 = ein, 0 = aus)

### 1.1.2.9 Rückgabecodes der MF II-Tastatur

Mit den in diesem Abschnitt dargestellten Rückgabecodes [16] reagiert die Tastatur auf die gesendeten Befehle.

Code	Bedeutung
00h	Überlauffehler
FFh	Tastenfehler
41ABh	Tastatur-ID für eine MF II-Tastatur
AAh	BAT-Abschlusscode
EEh	Echo nach einem Echo-Befehl
FAh	ACK ( <b>A</b> cknowledge)
FCh	BAT-Fehler
FEh	Resend-Anforderung

**Tabelle 5.** Rückgabecodes der MF II-Tastatur

### 1.1.3 Versuch Nr. 1

**Aufgabe:** Aufzeichnung des Protokolls der Tastatur nach C167-Übertragung mit einem Digitalspeicheroszilloskop (DSO)

**Lernziele:**

- Vertiefung der im Abschnitt 1.1.2.4 dargelegten Grundlagen
- Bedienung eines modernen Digitalspeicheroszilloskops

**Anmerkungen:** Es wird davon ausgegangen, dass das DSO TDS 3034 der Firma Tektronix verwendet wird.  
Verwenden Sie nur die zum DSO gehörenden Tastköpfe (Tek P6139A).  
Bei Fragen zum Digitalspeicheroszilloskop sollten Sie im Handbuch nachschlagen.

#### **Durchführung:**

- (1) Kanal 1 (CH 1) des DSO über Tastkopf mit TP67 (CLK) auf der LCD-Keyboard-I/O-Platine verbinden.  
Kanal 2 (CH 2) des DSO über Tastkopf mit TP64 (DATA) auf der LCD-Keyboard-I/O-Platine verbinden.  
Die Krokodilklemmen sind an den Massepunkten (schwarz) anzuschließen (siehe auch Abb.4).
- (2) Stellen Sie am Bildschirm des DSO die beiden Kanäle dar.  
Da die Einstellmenüs des verwendeten DSO selbsterklärend sind, werden sie in dieser Anleitung nur am Rande erläutert. Für die Einstellungen am DSO sollten Sie sich ruhig etwas Zeit lassen und die einzelnen Menüs einmal am Oszilloskopbildschirm darstellen.

Empfohlene Einstellungen: CH 1 -> 2.00 V/Div

CH 2 -> 2.00 V/Div

Zeitachse -> 100  $\mu$ s/Div

Trigger -> CH 1, fallende Flanke, Level = 1.00 V

Zum Aktivieren der Triggerfunktion müssen Sie noch die „SINGLE SEQ“-Taste betätigen, wodurch die entsprechende LED am Oszilloskop aufleuchtet.

- (3) Drücken Sie eine Zifferntaste auf dem Hauptblock der Tastatur. Im weiteren wird davon ausgegangen, dass Sie die Zifferntaste „1“ betätigt haben.
- (4) Verfolgen Sie die Ausgaben am Bildschirm des DSOs.
- (5) Speichern Sie den Bildschirminhalt des Oszilloskops auf einer 3,5 Zoll-Diskette ab (im PCX-Format). Hierzu muss die „Drucker“-Taste am DSO betätigt werden. Öffnen Sie die Datei am Host-Rechner (z.B. mit Microsoft Photo Editor) und lassen Sie die Farben invertieren. Zum Schluss müssen Sie die Datei noch ausdrucken.
- (6) Vergleichen Sie den Ausdruck mit Abb. 7 und kennzeichnen Sie die einzelnen Bits auf der DATA-Leitung (CH 2 am Oszilloskop).

**Hinweis:** Versuch Nr. 1 nicht abbauen, da er als Grundlage für den Versuch Nr. 2 verwendet wird.

Hiermit ist der Versuch Nr. 1 beendet.

### 1.1.4 Versuch Nr. 2

**Aufgabe:** Aufzeichnung der Make- und Break-Codes einer MF II-Tastatur mit dem Digitalspeicheroszilloskop.

**Lernziele:**

- Vertiefung der im Abschnitt 1.1.2.5 vermittelten Kenntnisse
- Bedienung eines modernen Digitalspeicheroszilloskops

**Anmerkungen:** Siehe Versuch Nr. 1 (Abschnitt 1.1.3)

#### **Durchführung:**

- (1) Vergrößern Sie am DSO die Zeitachse auf 10.0 ms/Div.
- (2) Drücken Sie die Zifferntaste „1“ auf dem Hauptblock der Tastatur.  
Am DSO sollten nun drei vertikale Linien dargestellt werden.
- (3) Vergrößern Sie die linke vertikale Linie, indem Sie den Zoom-Modus des DSO benutzen, der mit der Taste „Lupe“ aufgerufen wird. Nachdem die Taste „Lupe“ betätigt wurde, leuchtet die entsprechende LED am DSO auf.
- (4) Verschieben Sie nun den Zoom-Bereich in horizontaler Richtung, bis der gewünschte Ausschnitt in der Mitte des DSO-Bildschirms dargestellt wird.
- (5) Verringern Sie den Maßstab der Zeitachse auf 100  $\mu$ s/Div, um den dargestellten Ausschnitt besser analysieren zu können.
- (6) Die Ausgabe des DSO ist nun identisch mit dem Ausdruck aus Versuch Nr. 1 (Abschnitt 1.1.3) und stellt den sogenannten Make-Code dar.
- (7) Vergrößern Sie nun die mittlere vertikale Linie.



- (8) Speichern Sie den Bildschirminhalt des Oszilloskops auf einer 3,5 Zoll-Diskette ab (im PCX-Format). Hierzu muss die „Drucker“-Taste am DSO betätigt werden. Öffnen Sie die Datei am Host-Rechner (z.B. mit Microsoft Photo Editor) und lassen Sie die Farben invertieren. Zum Schluss müssen Sie die Datei noch ausdrucken.
- (9) Markieren Sie auf dem Ausdruck die einzelnen Bits der DATA-Leitung (CH 2 am Oszilloskop). Die acht Datenbits auf der DATA-Leitung entsprechen dem Break-Code (F0h), der beim Loslassen einer Taste erzeugt wird.
- (10) Zur Analyse der rechten vertikalen Linie ist der Zoom-Bereich nach rechts zu verschieben.
- (11) Speichern und drucken Sie den Bildschirminhalt des Oszilloskops.
- (12) Markieren Sie auf dem Ausdruck die einzelnen Bits der DATA-Leitung (CH2 am Oszilloskop). Die acht Datenbits stellen wie schon im Schritt (6) den Make-Code der gedrückten bzw. losgelassenen Taste dar.
- (13) Verlassen Sie den Zoom-Modus des DSO, indem Sie die „Lupe“-Taste erneut betätigen. Die LED am DSO erlischt daraufhin wieder.
- (14) Speichern und drucken Sie den Bildschirminhalt des Oszilloskops.

Hiermit ist der Versuch Nr. 2 beendet.

### 1.1.5 Versuch Nr. 3

**Aufgabe:** Ziffern 0...9 im Polling-Betrieb von der Tastatur einlesen.

**Lernziele:**

- Vertiefung der im Abschnitt 1.1.2.4 vermittelten Kenntnisse
- Vertiefung der Grundlagen aus der Mikrocomputervorlesung
- Anwendung des Polling-Betriebs

**Anmerkung:** An dieser Stelle wird nicht auf die Bedienung der UltraEdit 32-Entwicklungsumgebung eingegangen.

**Durchführung:**

- (1) Kodieren Sie das Hauptprogramm „read\_from\_keyboard“.  
Den zugehörigen Programmablaufplan finden Sie in Anlage 1 (Abb. 11).  
Ein Programmfragment ist in Anlage 4 (Seite 23) dargestellt.
- (2) Kodieren Sie das lokale Unterprogramm „rdkey“.  
Den zugehörigen Programmablaufplan finden Sie in Anlage 2 (Abb. 12).
- (3) Kodieren Sie das lokale Unterprogramm „scanCode2ascii“.  
Den zugehörigen Programmablaufplan finden Sie in Anlage 3 (Abb. 13).
- (4) Erstellen Sie die Applikation mit der UltraEdit 32-Entwicklungsumgebung und transferieren Sie das erzeugte Ladeprogramm (Endung .hex) in das RAM des Mikrocontrollers SAB-C167CR-LM.  
Starten Sie die Anwendung.

- (5) Testen Sie das Programm, mit den Debuggerfunktionen des Monitors.  
Drücken Sie nacheinander die Zifferntasten 0...9 auf dem Hauptblock der Tastatur. Halten Sie die Tasten auch mal längere Zeit gedrückt.  
Wenn die Anwendung funktioniert, müssen Sie noch den Quellcode ausdrucken.

**Hinweis:** Die lokalen Unterprogramme „scanCode2ascii“ und „rdkey“ werden im Versuch Nr. 4 (Abschnitt 1.1.6) bzw. Versuch Nr. 5 (Abschnitt 1.1.7) noch mal benötigt.

Hiermit ist der Versuch Nr. 3 beendet.

### 1.1.5.1 Material zur Versuchsdurchführung

- Anlage 1: Programmablaufplan für Hauptprogramm „read\_from\_keyboard“<sup>1</sup>

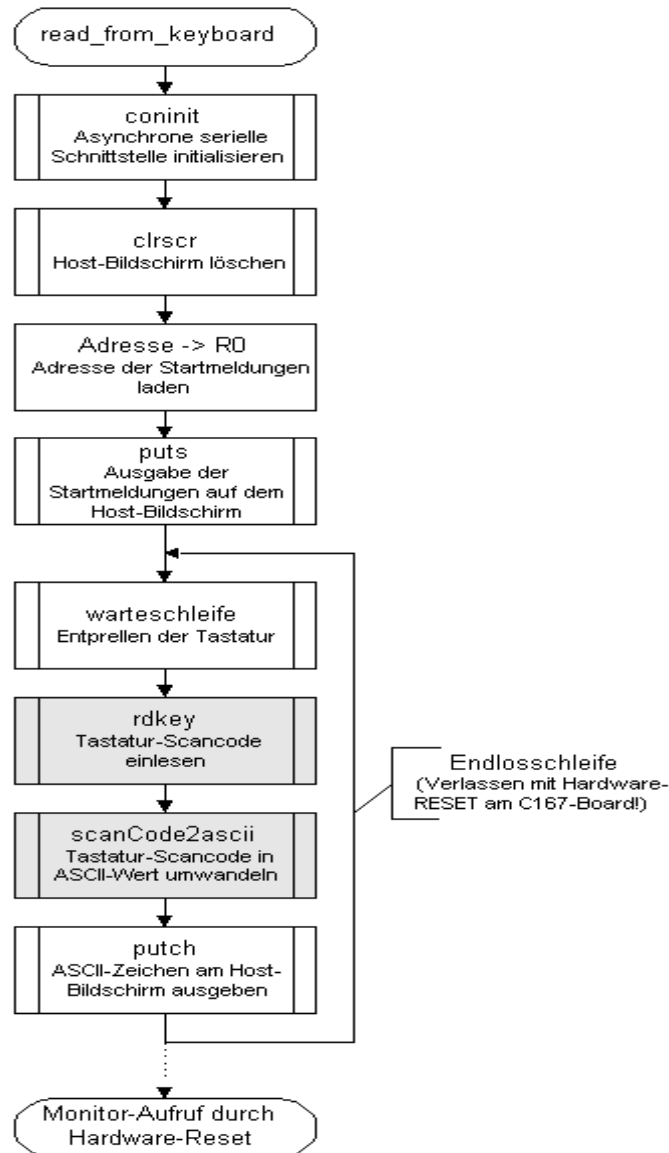


Abb. 11. Programmablaufplan für Hauptprogramm „read\_from\_keyboard“

<sup>1</sup> Die Unterprogramme „coninit“, „clrscr“, „putch“, „puts“ und „warteschleife“ können mit %include „UP\_ser.ASM“ dem eigenen Programm hinzugefügt werden.

• Anlage 2: Programmablaufplan für lokales Unterprogramm „rdkey“

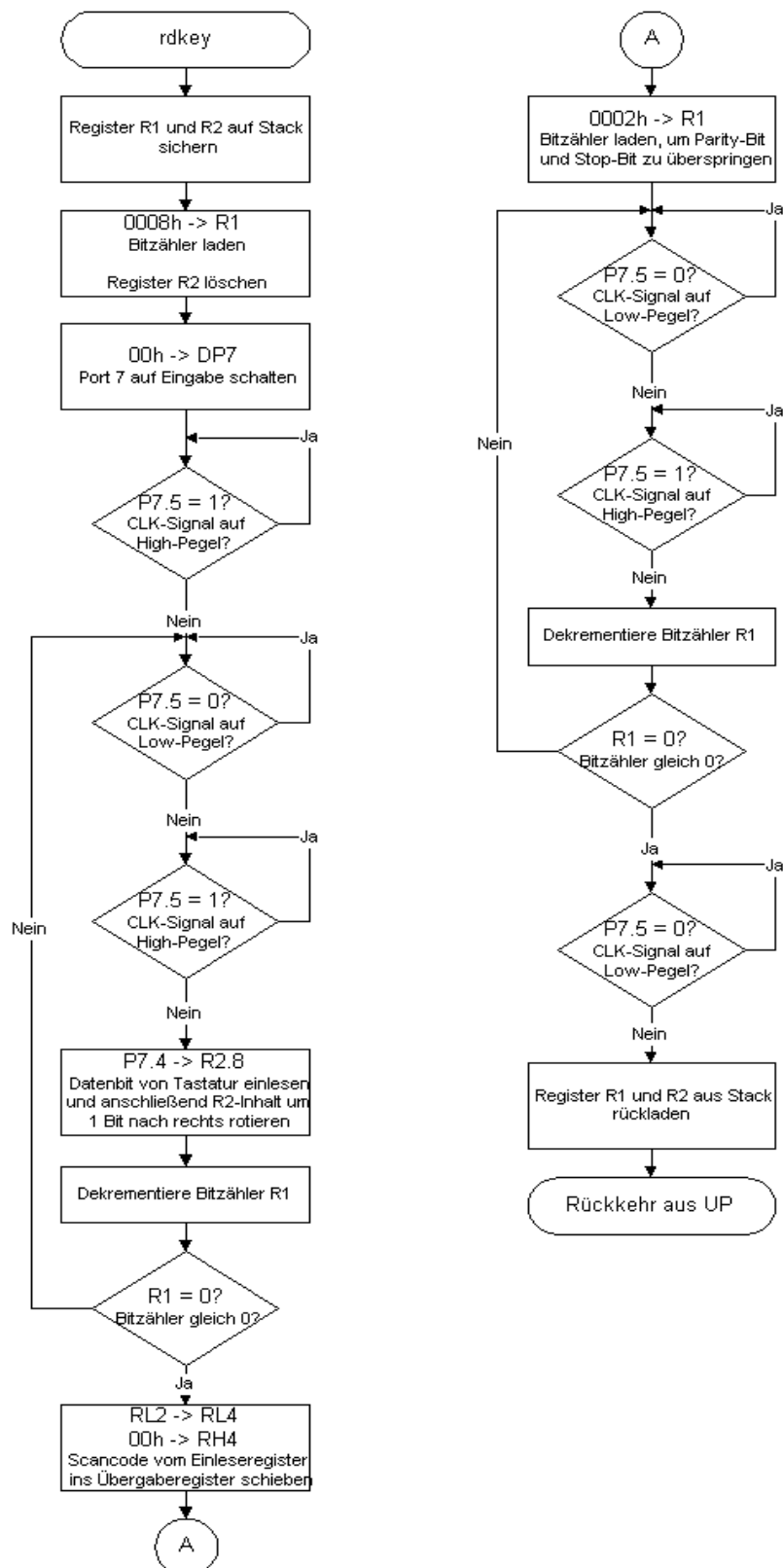


Abb. 12. Programmablaufplan für lokales Unterprogramm „rdkey“

- Anlage 3: Programmablaufplan für lokales Unterprogramm „scanCode2ascii“

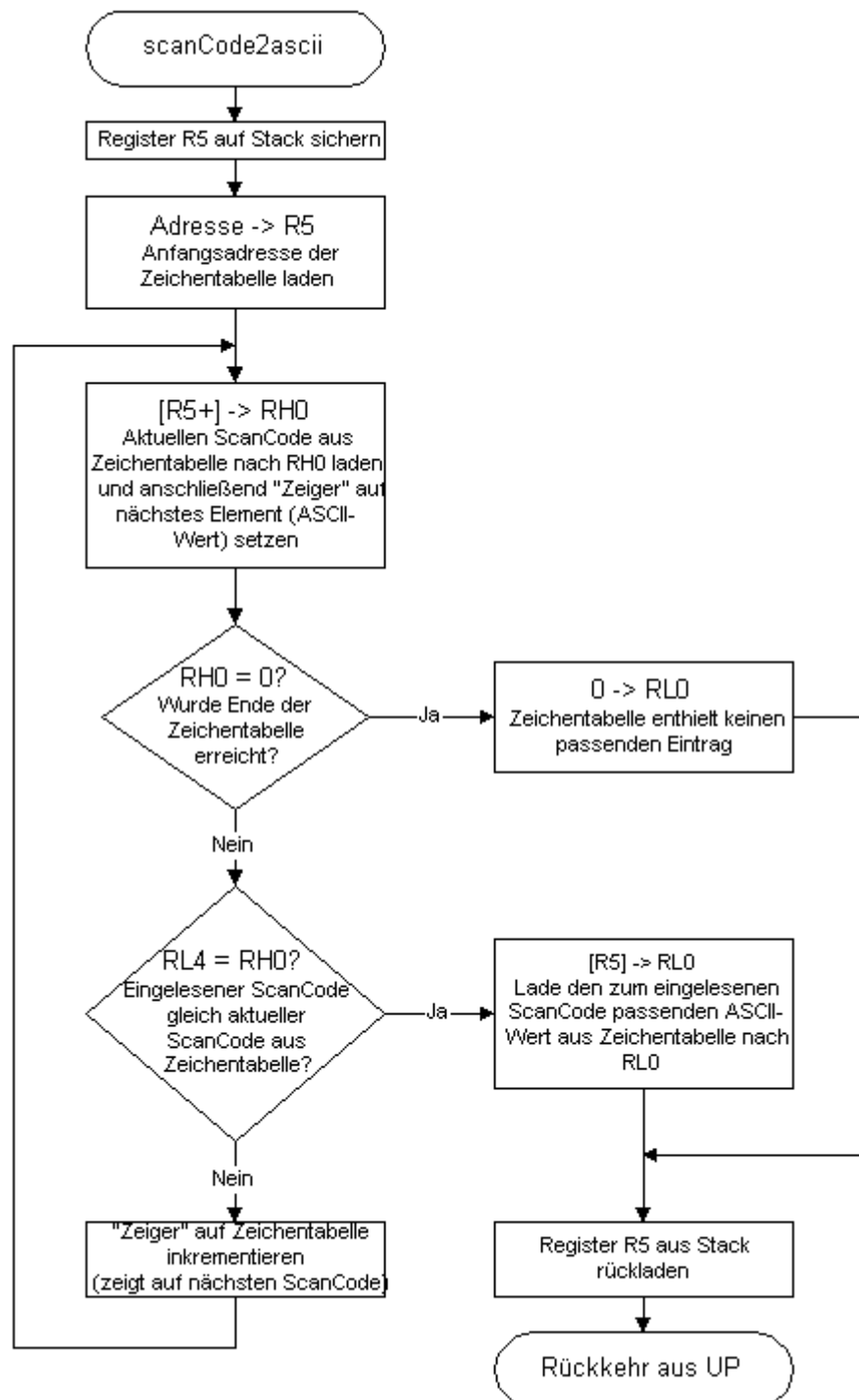


Abb. 13. Programmablaufplan für lokales Unterprogramm „scanCode2ascii“

- Anlage 4: Programmfragment „read\_from\_keyboard“

```

; -----
; Konstantendefinitionen (Equates)
; -----
counter_read      EQU 0008h          ;8 Datenbits
par_stop_count    EQU 0002h          ;1 Paritybit + 1 Stopbit
null              EQU 0000h          ;Vorbelegung mit Null
portIn            EQU 00h            ;Port auf Eingabe setzen

; -----
; Datenbereich
; -----
; ---- Startmeldung ----
Start  DB lf,cr,lf,cr,'-----'
        DB lf,cr,'Entwicklungsumgebung zum Programmieren des C167-Boards in Assembler'
        DB lf,cr,lf,cr,lf,cr,'Programm: read_from_keyboard.ASM'
        DB lf,cr,'          (Ziffern 0...9 im Polling-Betrieb von der Tastatur einlesen)'
        DB lf,cr,'-----'
        DB lf,cr,lf,cr,'Gedruckte Taste: ',0

; ---- Zeichentabelle mit Scancodes und zugehoerigen ASCII-Zeichen ----
; ---- Umwandlungstabelle fuer Ziffern 0...9 des Hauptblocks der Tastatur ----
zeichen_tabelle  DB 045h,'0',016h,'1',01Eh,'2',026h,'3',025h,'4'
                  DB 02Eh,'5',036h,'6',03Dh,'7',03Eh,'8',046h,'9',0

```

### 1.1.6 Versuch Nr. 4

**Aufgabe:** Ziffern 0...9 im Interrupt-Betrieb von der Tastatur einlesen

**Lernziele:**

- Vertiefung der im Abschnitt 1.1.2.4 vermittelten Kenntnisse
- Vertiefung der Grundlagen aus der Mikrocomputervorlesung
- Anwendung des Interrupt-Betriebs

**Anmerkungen:** Bei einer fallenden Signalfanke an P7.5 (CLK) soll die Interrupt-Service-Routine (ISR) „rdkey\_int“ aufgerufen werden, die die einzelnen Bits von P7.4 (DATA) einliest.  
Das Programm soll mit dem Schalter an P8.0 beendet werden können.

**Durchführung:**

- (1) Kodieren Sie das Hauptprogramm „read\_with\_interrupt“.  
Zum Einbinden der Unterprogrammbibliothek „UP\_ser.ASM“ siehe Seite 20.  
Den zugehörigen Programmablaufplan finden Sie in Anlage 5 (Abb. 14).  
Ein Programmfragment ist in Anlage 7 (Seite 28) dargestellt.
- (2) Kodieren Sie die Interrupt-Service-Routine „rdkey\_int“.  
Den zugehörigen Programmablaufplan finden Sie in Anlage 6 (Abb. 15).  
Das Unterprogramm „scanCode2ascii“ wurde schon im Versuch Nr. 3 (Abschnitt 1.1.5) erstellt und soll hier wiederverwendet werden.  
Zur Anfertigung dieser ISR siehe auch Abb. 7.



- (3) Erstellen und Testen Sie die Applikation mit der UltraEdit 32-Entwicklungsumgebung.

Drücken Sie nacheinander die Zifferntasten 0...9 auf dem Hauptblock der Tastatur. Halten Sie die Tasten auch mal längere Zeit gedrückt.

Wenn die Anwendung funktioniert, müssen Sie noch den Quellcode ausdrucken.

Hiermit ist der Versuch Nr. 4 beendet.

### 1.1.6.1 Material zur Versuchsdurchführung

- Anlage 5: Programmablaufplan für Hauptprogramm „read\_with\_interrupt“

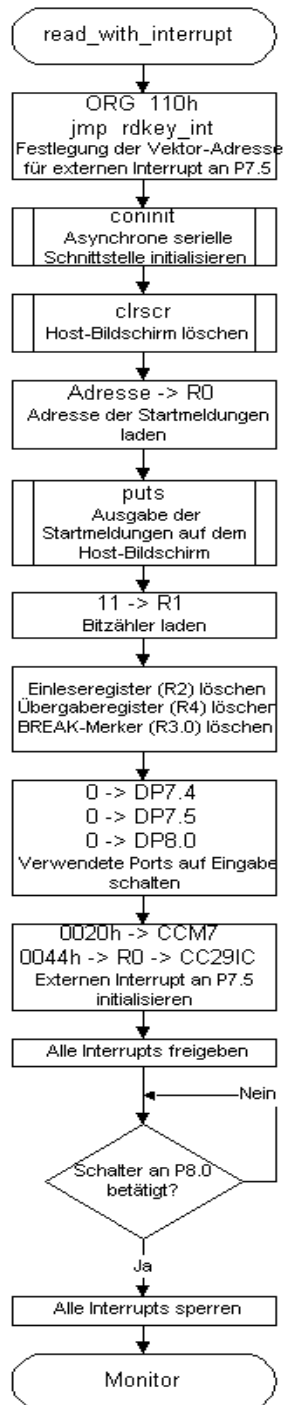


Abb. 14. Programmablaufplan für Hauptprogramm „read\_with\_interrupt“

- Anlage 6: Programmablaufplan für ISR „rdkey\_int“

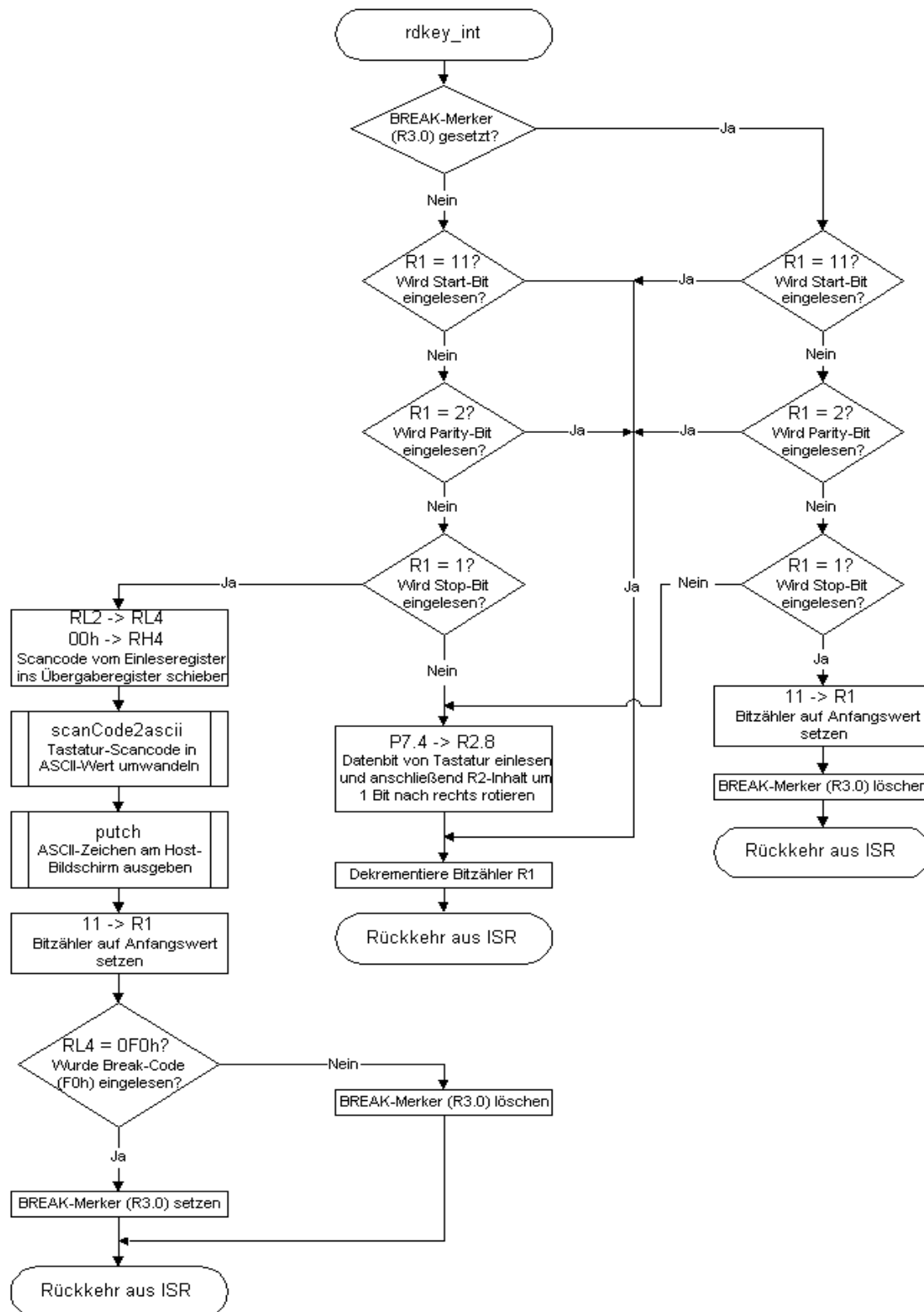


Abb. 15. Programmablaufplan für ISR „rdkey\_int“

- Anlage 7: Programmfragment „read\_with\_interrupt“

```

; -----
; Konstantendefinitionen (Equates)
; -----
CC29IC      EQU 0F184h      ;Capture/Compare-Interrupt-Steuerung
counter_int EQU 000Bh      ;Zeichenzähler fuer Interrupt-Betrieb
                                ;(1 Startbit + 8 Datenbits + 1 Paritybit + 1 Stopbit)
init_CCM7   EQU 0020h      ;externer Interrupt an P7.5 mit fallender Flanke
init_CC29IC EQU 44h        ;Group-Level, Interrupt-Level, Interrupt an P7.5 freigeben
null        EQU 0000h      ;Vorbelegung mit Null

; -----
;Datenbereich
; -----
;---- Startmeldung ----
Start  DB lf,cr,lf,cr,'-----'
        DB lf,cr,'Entwicklungsumgebung zum Programmieren des C167-Boards in Assembler'
        DB lf,cr,lf,cr,lf,cr,'Programm: read_with_interrupt.ASM'
        DB lf,cr,'          (Ziffern 0...9 im Interrupt-Betrieb von der Tastatur einlesen)'
        DB lf,cr,lf,cr,'          Zum Beenden des Programmes Schalter an P8.0 betätigen!'
        DB lf,cr,'-----'
        DB lf,cr,lf,cr,'Gedruckte Taste: ',0

; ---- Zeichentabelle mit Scancodes und zugehoerigen ASCII-Zeichen ----
; ---- Umwandlungstabelle fuer Ziffern 0...9 des Hauptblocks der Tastatur ----
zeichen_tabelle DB 045h,'0',016h,'1',01Eh,'2',026h,'3',025h,'4'
                  DB 02Eh,'5',036h,'6',03Dh,'7',03Eh,'8',046h,'9',0

```

### 1.1.7 Versuch Nr. 5

**Aufgabe:** Senden von Befehlen an die MF II-Tastatur

**Lernziele:**

- Vertiefung der im Abschnitt 1.1.2.6 vermittelten Kenntnisse
- Vertiefung der Grundlagen aus der Mikrocomputervorlesung

**Anmerkung:** Es sollen die drei Tastatur-LEDs eingeschaltet werden.

**Durchführung:**

- (1) Das Hauptprogramm „send\_to\_keyboard“ wird für diesen Versuch bereits vorgegeben (C:\C167\Praktikum\Tastatur\send\send\_to\_keyboard.ASM).  
Der zugehörige Programmablaufplan ist in Anlage 8 dargestellt.  
Das Unterprogramm „rdkey“ wurde im Versuch Nr. 3 (Abschnitt 1.1.5) erstellt und soll hier wiederverwendet werden.
- (2) Kodieren Sie das lokale Unterprogramm „send2key“.  
Den zugehörigen Programmablaufplan finden Sie in Anlage 9 (Abb.17).  
Zur Anfertigung des Un.
- (3) Erstellen und Testen Sie die Anwendung mit der UltraEdit 32-Entwicklungsumgebung.  
Schalten Sie die unterschiedlichen Tastatur-LEDs an und aus.  
Senden Sie auch mal ein fehlerhaftes Indikatorbyte an die Tastatur.  
Wenn die Anwendung funktioniert, müssen Sie noch den Quellcode des lokalen Unterprogramms „send2key“ ausdrucken.

Hiermit ist der Versuch Nr. 5 beendet.



• Anlage 9: Programmablaufplan des lokalen Unterprogramms „send2key“

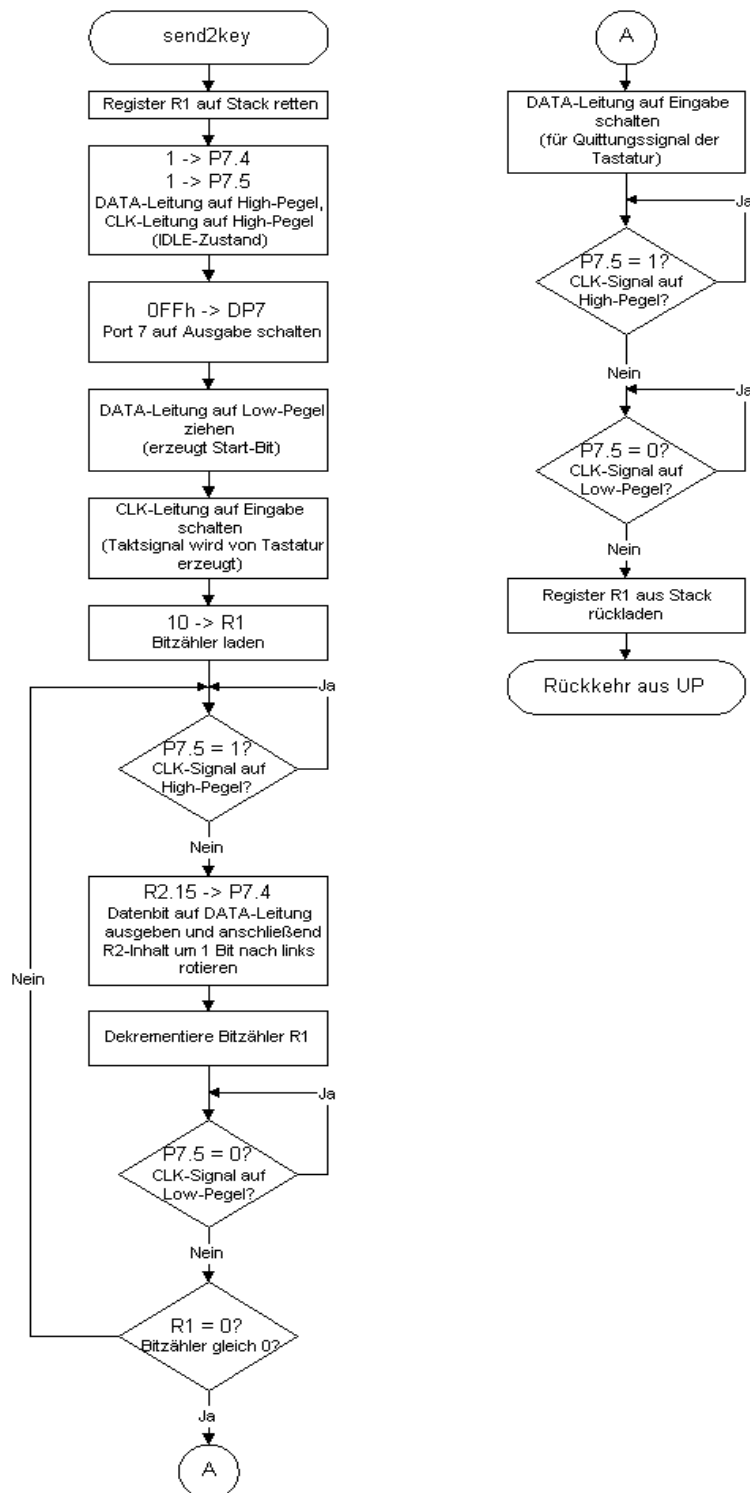


Abb.17. Programmablaufplan für lokales Unterprogramm „send2key“

## Literaturverzeichnis

- [1] Infineon technologies; C167 Derivatives User's Manual, Version 2.0, March 96
- [2] <http://www.infineon.com>
- [3] <http://www.phytec.de>
- [4] Phytec Messtechnik GmbH; miniMODUL-167 Hardwaremanual, Oktober 1999
- [5] Phytec Messtechnik GmbH; KitCON-167 Hardware-Manual, Version 3.1 12/1999
- [6] Phytec Messtechnik GmbH; Basisplatine für CAN und mini-/microMODULE Hardwaremanual, Februar 2000
- [7] [http://rfhs0005.fh-regensburg.de/~scg39398/ie\\_lab/Default.htm](http://rfhs0005.fh-regensburg.de/~scg39398/ie_lab/Default.htm)
- [8] ertec GmbH; ASS16X Optimizing Absolute Cross Assembler for SABC163/SABC165/SABC166/SABC167 Manual, Version 1, 1995
- [9] <http://www.ultraedit.com>
- [10] G. Schmitt; Mikrocomputertechnik mit dem Controller C167, Oldenbourg Wissenschaftsverlag GmbH, München 2000
- [11] Infineon technologies; SK-167/167CS Starter Kit Getting Started, Version 3.0 08/1999
- [12] <http://www.beyondlogic.org/keyboard/keybrd.htm>
- [13] Cherry GmbH; Model 1000/3000/6000 User Manual, Auerbach/Opf. 1999
- [14] Hans-Peter Messmer; PC Hardware Buch, 5. Auflage, Addison Wesley Longman GmbH, Bonn 1998
- [15] Infineon technologies; Instruction Set Manual for the C166 Family, Version 1.2, 12.97
- [16] W. Schindler; Skriptum zur Vorlesung Mikrocomputertechnik, FH Amberg-Weiden
- [17] Jörg Wiegelmann; Softwareentwicklung in C für Mikroprozessoren, Hüthig GmbH, Heidelberg 1996
- [18] Rolf Klaus; Der Mikrokontroller C167, vdf Hochschulverlag AG, 2000