
Eine einfache Lösung für Programmauswahl

Gunther Zielosko

1. Die Ausgangssituation

Vielleicht haben sich manche BASIC-Tiger® -Nutzer schon über die umfangreiche Tastatur des Plug-and-Play-Labs gewundert. Die Entwickler des Systems haben damit und mit den aufwendigen Maßnahmen bezüglich der Nutzung des LCD-Displays (Sonderzeichensätze, Cursorsteuerung usw.) dem PC ein bißchen Konkurrenz gemacht. Oft aber liegt die Stärke eines Mikrocontrollers gar nicht im Schreiben oder Darstellen von Text, sondern eher in einer zugeschnittenen Lösung eines ganz speziellen Problems. Benötigt man nur wenige Eingriffsmöglichkeiten in den Programmablauf, ist die Hard- und Software dagegen relativ einfach. Am Beispiel einer Minimalatastatur mit Programmauswahlschalter soll dieses Konzept erläutert werden.

2. Problemstellung

Als Elektroniker hat man viele kleine Probleme, die sich mit einem Mikrorechner wie dem BASIC-Tiger® elegant lösen lassen (Spannung messen, Impulslänge messen, Takt ausgeben, Bitmuster ausgeben, Daten an den PC senden usw.). Im folgenden wollen wir in einer Erweiterung der Arbeitsmöglichkeiten eine Programmsammlung anlegen, welche die Auswahl der Programme optimiert.

3. Hardware

Grundlage ist ein binärer Drehwahlschalter (siehe Bild 1), der je nach Ausführung bis zu 16 Stellungen hat und auf der Frontseite die entsprechende Ziffer anzeigt.



Bild 1 typischer Drehwahlschalter

Ausgangsseitig befinden sich 5 Lötstützpunkte, die z.B. mit:

Com	gemeinsamer Anschluß
1	Bit 2 ⁰
2	Bit 2 ¹
4	Bit 2 ²
8	Bit 2 ³

bezeichnet sind.

Der erfahrene Elektroniker weiß, wie es weitergeht. Legt man an den gemeinsamen Anschluß Vcc, erscheint an den Ausgängen je nach Schalterstellung das binäre Muster der jeweilig eingestellten Ziffer oder Zahl. Diese ist von einem Halb-Port des BASIC-Tigers® sofort zu lesen. Für die konkrete Aufgabe, die hier bearbeitet wird, legt man den gemeinsamen Anschluss an die Masse, damit das binäre Wort in „negativer Logik“ ausgegeben wird. Dann wird:

0 dezimal	0000 binär	1111 bitweise negiert
1	0001	1110
:	:	:
7	0111	1000
8	1000	0111

usw.

Wenn man einen Drehschalter mit 10 Stellungen hat, kann man 10 verschiedene Einstellungen an einem halben Port (hier 80 bis 83) realisieren.

2 andere Portleitungen (hier 70 und 71) werden dann direkt mit Tasten belegt, auch hier wieder ganz einfach gegen Masse. In geeigneter Weise reagiert die Software auf die Wahlschalterstellung sowie auf einzelne gedrückte Tasten.

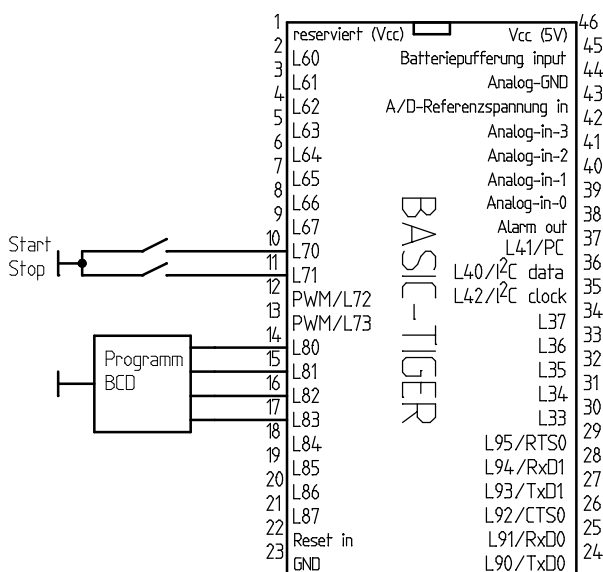


Bild 2 Schaltung des Programm-Wahlschalters

4. Software

Nehmen wir an, wir hätten die 2 Tasten mit den Funktionen "Start" und "Stop" belegt und wollen mit dem Wahlschalter "Programm" mit 10 Stellungen 10 verschiedene Programme auswählen, dann könnte unser Programme wie folgt funktionieren:

- Abfrage der Tasten
- Wenn Taste 1 gedrückt, Abfrage des Wahlschalters
- Ausgabe „Programm x“ (dies dient jetzt nur zur Kontrolle, kann später entfallen)
- Sprung zum dazugehörigen Unterprogramm (das z.Zt. noch ohne Funktion ist!)
- Wenn das beendet wurde (END erreicht) oder die Taste 2 gedrückt wurde
- Rücksprung zum Hauptprogramm, Ausgabe „Ende“, wiederum Tastenabfrage usw.

Genau das macht das folgende Programm. Mit dieser einfachen Hardware und wenig Cursor- und Menü- Akrobatik kann man auf 10 und mehr verschiedene Funktionen umschalten, Ziffern eingeben usw. Dank der vorgestellten Methode kann bei bestimmten Aufgaben sogar ganz auf ein Display (wer hat davon schon genug?) verzichtet werden, der Eingabeschalter hat ja bereits ein „Display“. Als beispielhafte Anwendung wäre ein Funktionsgenerator zu nennen, jede Funktion (Sinus, Rechteck, Dreieck, Sägezahn usw.) hat ihre Nummer, die mit dem Wahlschalter vorgewählt und mit „Start“ aktiviert wird. Mit „Stop“ wird diese Funktion beendet und ggf. danach eine neue gestartet. Viel Spaß beim Experimentieren!

```
-----
' Name: PRO_UM1.TIG           Beispiel für Umschaltung von 10 Programmen
-----

BYTE Programm                ' enthaelt Programm-Nummer
BYTE Taste                   ' enthaelt Tasten-Nummern

-----
TASK MAIN                    ' Beginn Task MAIN (Hauptprogramm)

INSTALL DEVICE #1, "LCD1.TDD" ' LCD installieren
Taste = 0                    ' Zustaende der beiden Tasten und
Programm = 0                  ' Wahlschalter Programm und

Einlesen:
Taste = 0                    ' Taste auf 0 setzen
CALL Tasten                  ' Tasten abfragen (Port 7)
WHILE Taste = 0              ' solange keine Taste gedruickt
CALL Tasten                  ' etwas warten und Tasten wieder
ENDWHILE                     ' wenn Taste gedruickt,
CALL Auswahl                 ' Auswahlshalter abfragen
IF Taste = 1 THEN
SWITCH Programm              ' Programmauswahl
CASE 0:
PRINT #1, "<1>Programm 0"
CALL Programm0               ' Programm 0
PRINT #1, "Ende"
GOTO Einlesen
WAIT_DURATION 300
CASE 1:
PRINT #1, "<1>Programm 1"
CALL Programm1               ' Programm 1
PRINT #1, "Ende"
GOTO Einlesen
```

```
        WAIT_DURATION 300
CASE 2:
    PRINT #1, "<1>Programm 2"
    CALL Programm2           ' Programm 2
    PRINT #1, "Ende"
    GOTO Einlesen
    WAIT_DURATION 300
CASE 3:
    PRINT #1, "<1>Programm 3"
    CALL Programm3           ' Programm 3
    PRINT #1, "Ende"
    GOTO Einlesen
    WAIT_DURATION 300
CASE 4:
    PRINT #1, "<1>Programm 4"
    CALL Programm4           ' Programm 4
    PRINT #1, "Ende"
    GOTO Einlesen
    WAIT_DURATION 300
CASE 5:
    PRINT #1, "<1>Programm 5"
    CALL Programm5           ' Programm 5
    PRINT #1, "Ende"
    GOTO Einlesen
    WAIT_DURATION 300
CASE 6:
    PRINT #1, "<1>Programm 6"
    CALL Programm6           ' Programm 6
    PRINT #1, "Ende"
    GOTO Einlesen
    WAIT_DURATION 300
CASE 7:
    PRINT #1, "<1>Programm 7"
    CALL Programm7           ' Programm 7
    PRINT #1, "Ende"
    GOTO Einlesen
    WAIT_DURATION 300
CASE 8:
    PRINT #1, "<1>Programm 8"
    CALL Programm8           ' Programm 8
    PRINT #1, "Ende"
    GOTO Einlesen
    WAIT_DURATION 300
CASE 9:
    PRINT #1, "<1>Programm 9"
    CALL Programm9           ' Programm 9
    PRINT #1, "Ende"
    GOTO Einlesen
    WAIT_DURATION 300
ENDSWITCH
ENDIF

GOTO Einlesen           ' nochmal einlesen

END
'-----
'-----
' Unterprogramm fragt Port 8 (Bits 0-3) auf eingestellte Nummernschalter
' fuer die Programm-Nr ab. Der Nummernschalter ist aktiv gegen Masse,
' deshalb werden die Ergebnisse negiert.
'-----
SUB Auswahl           ' Unterprogramm Switch und Taste

    BYTE Eingang           ' Hilfsvariable Eingang

    DIR_PORT 8,255           ' Port 8 als Eingang setzen
    IN 8,Eingang           ' Port einlesen (Programm, Speicher)
    Programm = BITNOT (Eingang) ' Programm-Nr. aus unteren Bits bilden
```

```

        Programm = Programm BITAND 00001111b ' negieren und obere Bits ausblenden
END                                           ' Ende Unterprogramm
'-----
'-----
' Unterprogramm fragt Port 7 (Bits 0 und 1) auf gedruckte Tasten ab.
' Die Tasten sind aktiv gegen Masse, deshalb werden die Ergebnisse
' negiert.
'-----
SUB Tasten                                  ' Unterprogramm Tastenabfrage

    DIR_PORT 7,255                          ' Port 7 als Eingang setzen
    IN 7,Taste                              ' Taste einlesen (Execute,)
    Taste = BITNOT (Taste)                  ' alle Bits von Taste negieren
    Taste = Taste BITAND 00000011b         ' 2 Bits abfragen (Port 7 hat nur 2 Bits)
END                                           ' Ende Unterprogramm
'-----

'
'          ab hier kommen die aufzurufenden Teilprogramme
'-----
' Unterprogramm0
'-----
SUB Programm0                              ' Das Programm ist (noch) leer
Anfang0:

CALL Tasten                                ' hier Programminhalt
IF Taste = 2 THEN                          ' Tasten abfragen
    GOTO Ende0                              ' Wenn Stoptaste gedrückt
ELSE
    GOTO Anfang0                            ' Wenn nicht, weiter
ENDIF
Ende0:
END
'-----
'-----
' Unterprogramm1
'-----
SUB Programm1                              ' Das Programm ist (noch) leer
Anfang1:

CALL Tasten                                ' hier Programminhalt
IF Taste = 2 THEN                          ' Tasten abfragen
    GOTO Endel                              ' Wenn Stoptaste gedrückt
ELSE
    GOTO Anfang1                            ' Wenn nicht, weiter
ENDIF
Endel:
END
'-----
'-----
' Unterprogramm0
'-----
SUB Programm2                              ' Das Programm ist (noch) leer
Anfang2:

CALL Tasten                                ' hier Programminhalt
IF Taste = 2 THEN                          ' Tasten abfragen
    GOTO Ende2                              ' Wenn Stoptaste gedrückt
ELSE
    GOTO Anfang2                            ' Wenn nicht, weiter
ENDIF
```

```
Ende2:
END
'-----
'
'-----
' Unterprogramm3
'-----
SUB Programm3                                ' Das Programm ist (noch) leer
Anfang3:                                     '
                                           ' hier Programminhalt
CALL Tasten                                  ' Tasten abfragen
IF Taste = 2 THEN
  GOTO Ende3                                 ' Wenn Stoptaste gedrückt
ELSE
  GOTO Anfang3                               ' Wenn nicht, weiter
ENDIF
Ende3:
END
'-----
'
'-----
' Unterprogramm4
'-----
SUB Programm4                                ' Das Programm ist (noch) leer
Anfang4:                                     '
                                           ' hier Programminhalt
CALL Tasten                                  ' Tasten abfragen
IF Taste = 2 THEN
  GOTO Ende4                                 ' Wenn Stoptaste gedrückt
ELSE
  GOTO Anfang4                               ' Wenn nicht, weiter
ENDIF
Ende4:
END
'-----
'
'-----
' Unterprogramm5
'-----
SUB Programm5                                ' Das Programm ist (noch) leer
Anfang5:                                     '
                                           ' hier Programminhalt
CALL Tasten                                  ' Tasten abfragen
IF Taste = 2 THEN
  GOTO Ende5                                 ' Wenn Stoptaste gedrückt
ELSE
  GOTO Anfang5                               ' Wenn nicht, weiter
ENDIF
Ende5:
END
'-----
'
'-----
' Unterprogramm6
'-----
SUB Programm6                                ' Das Programm ist (noch) leer
Anfang6:                                     '
                                           ' hier Programminhalt
CALL Tasten                                  ' Tasten abfragen
IF Taste = 2 THEN
  GOTO Ende6                                 ' Wenn Stoptaste gedrückt
ELSE
  GOTO Anfang6                               ' Wenn nicht, weiter
ENDIF
Ende6:
END
'-----
'
'-----
' Unterprogramm7
```

```
'-----  
SUB Programm7                                ' Das Programm ist (noch) leer  
Anfang7:                                     ' hier Programminhalt  
  
CALL Tasten                                 ' Tasten abfragen  
IF Taste = 2 THEN  
    GOTO Ende7                               ' Wenn Stoptaste gedrückt  
ELSE  
    GOTO Anfang7                             ' Wenn nicht, weiter  
ENDIF  
Ende7:  
END  
  
'-----  
  
' Unterprogramm8  
'-----  
SUB Programm8                                ' Das Programm ist (noch) leer  
Anfang8:                                     ' hier Programminhalt  
  
CALL Tasten                                 ' Tasten abfragen  
IF Taste = 2 THEN  
    GOTO Ende8                               ' Wenn Stoptaste gedrückt  
ELSE  
    GOTO Anfang8                             ' Wenn nicht, weiter  
ENDIF  
Ende8:  
END  
  
'-----  
  
' Unterprogramm9  
'-----  
SUB Programm9                                ' Das Programm ist (noch) leer  
Anfang9:                                     ' hier Programminhalt  
  
CALL Tasten                                 ' Tasten abfragen  
IF Taste = 2 THEN  
    GOTO Ende9                               ' Wenn Stoptaste gedrückt  
ELSE  
    GOTO Anfang9                             ' Wenn nicht, weiter  
ENDIF  
Ende9:  
END
```